

Manuel PHP 4.3.8 - Version Française

<http://www.nexen.net>

Sommaire

Manuel PHP	1
1.1 translators.....	1
1.2 Note légale.....	1
2 Préface	2
2.1 Auteurs et Contributeurs.....	2
3 Au moment de commencer	3
3.1 Introduction.....	3
3.1.1 Qu'est ce que PHP?.....	3
3.1.2 Que peut faire PHP?.....	3
3.2 Une introduction à PHP.....	5
3.2.1 Le nécessaire.....	6
3.2.2 Votre première page PHP.....	6
3.2.3 Trucs pratiques.....	8
3.2.4 Utiliser un formulaire.....	10
3.2.5 Utiliser des codes anciens avec les nouvelles versions de PHP.....	10
3.2.6 Et après?.....	11
4 Installation et configuration	12
4.1 Considérations générales sur l'installation.....	12
4.2 Installation sous UNIX.....	12
4.2.1 Apache 1.3.x sur les systèmes Unix.....	13
4.2.2 Apache 2.0 sur les systèmes Unix.....	17
4.2.3 Caudium.....	19
4.2.4 Installation avec les serveurs fhttpd.....	20
4.2.5 Installation sous Netscape et iPlanet Enterprise Serveur sur un système Sun Solaris.....	20
4.2.6 CGI et configurations en ligne de commande.....	23
4.2.7 Installation sous HP-UX.....	24
4.2.8 Installations sur les systèmes OpenBSD.....	25
4.2.9 Installation sous Solaris.....	27
4.2.10 Installation sous Gentoo.....	27
4.2.11 Installation sous Debian GNU/Linux.....	29
4.3 Installation sur un système Mac OS X.....	31
4.3.1 Utilisation des paquets.....	31
4.3.2 Compilation pour les serveurs OS X.....	31
4.3.3 Compilation pour les clients MacOs.....	32
4.4 Installation sur les système Windows.....	33
4.4.1 L'installateur Windows.....	33
4.4.2 Installation manuelle sous Windows.....	34
4.4.3 ActiveScript.....	38
4.4.4 Installation avec les serveurs IIS/PWS.....	39
4.4.5 Installer PHP sous Microsoft Windows avec Apache 1.3.x.....	42
4.4.6 Installation des serveurs Apache 2.0.x sur les systèmes Microsoft Windows.....	43
4.4.7 Serveurs Sun, iPlanet et Netscape servers sur Microsoft Windows.....	45
4.4.8 Installation pour les serveurs OmniHTTPd.....	48
4.4.9 Sambar Server on Microsoft Windows.....	49
4.4.10 Installation Xitami sur Microsoft Windows.....	49
4.4.11 Compilation des sources.....	50
4.4.12 Installation des extensions sous Windows.....	53
4.5 Installation d'extensions PECL.....	56

Sommaire

4 Installation et configuration

<u>4.5.1 Introduction aux installations PECL</u>	56
<u>4.5.2 Télécharger des extensions PECL</u>	57
<u>4.5.3 PECL sur Windows</u>	57
<u>4.5.4 Compilations d'extensions PECL partagées avec PEAR</u>	57
<u>4.5.5 Compilation des extensions partagées avec phpize</u>	58
<u>4.5.6 Compilation des extensions PECL statiquement dans PHP</u>	58
<u>4.6 Configuration</u>	59
<u>4.6.1 Le fichier de configuration</u>	59
<u>4.6.2 Comment modifier la configuration</u>	60

5 Référence du langage..... **63**

<u>5.1 La syntaxe de base</u>	63
<u>5.1.1 Passer du HTML au PHP</u>	63
<u>5.1.2 Séparation des instructions</u>	64
<u>5.1.3 Commentaires</u>	65
<u>5.2 Les types</u>	65
<u>5.2.1 Introduction</u>	65
<u>5.2.2 Booléens</u>	67
<u>5.2.3 Entiers</u>	68
<u>5.2.4 Les nombres décimaux</u>	71
<u>5.2.5 Les chaînes de caractères</u>	71
<u>5.2.6 Les tableaux</u>	79
<u>5.2.7 Les objets</u>	88
<u>5.2.8 Ressources</u>	89
<u>5.2.9 La valeur NULL</u>	89
<u>5.2.10 Pseudo-types utilisés dans cette documentation</u>	90
<u>5.2.11 Définition du type</u>	91
<u>5.3 Les variables</u>	93
<u>5.3.1 Essentiel</u>	93
<u>5.3.2 Variables pré-définies</u>	94
<u>5.3.3 Portée des variables</u>	96
<u>5.3.4 Les variables dynamiques</u>	100
<u>5.3.5 Variables externes à PHP</u>	101
<u>5.4 Les constantes</u>	104
<u>5.4.1 Syntaxe</u>	105
<u>5.4.2 Constantes magiques</u>	106
<u>5.5 Les expressions</u>	106
<u>5.6 Les opérateurs</u>	109
<u>5.6.1 La précedence des opérateurs</u>	109
<u>5.6.2 Les opérateurs arithmétiques</u>	110
<u>5.6.3 Les opérateurs d'assignation</u>	110
<u>5.6.4 Opérateurs sur les bits</u>	111
<u>5.6.5 Opérateurs de comparaison</u>	112
<u>5.6.6 Opérateur de contrôle d'erreur</u>	114
<u>5.6.7 Opérateur d'exécution</u>	114
<u>5.6.8 Opérateurs d'incrémentatation et décrémentatation</u>	115
<u>5.6.9 Les opérateurs logiques</u>	116
<u>5.6.10 Opérateurs de chaînes</u>	116
<u>5.6.11 Opérateurs de tableaux</u>	117
<u>5.6.12 Opérateur de type</u>	118
<u>5.7 Les structures de contrôle</u>	118

Sommaire

5 Référence du langage

<u>5.7.1 if</u>	118
<u>5.7.2 else</u>	119
<u>5.7.3 elseif</u>	120
<u>5.7.4 Syntaxe alternative</u>	120
<u>5.7.5 while</u>	121
<u>5.7.6 do-while</u>	121
<u>5.7.7 for</u>	122
<u>5.7.8 foreach</u>	123
<u>5.7.9 break</u>	126
<u>5.7.10 continue</u>	126
<u>5.7.11 switch</u>	127
<u>5.7.12 declare</u>	130
<u>5.7.13 return</u>	131
<u>5.7.14 require</u>	132
<u>5.7.15 include</u>	132
<u>5.7.16 require_once</u>	136
<u>5.7.17 include_once</u>	137
<u>5.8 Les fonctions</u>	138
<u>5.8.1 Les fonctions</u>	138
<u>5.8.2 Les arguments de fonction</u>	140
<u>5.8.3 Les valeurs de retour</u>	142
<u>5.8.4 Fonctions variables</u>	143
<u>5.8.5 Fonctions internes</u>	144
<u>5.9 Les classes et les objets (PHP 4)</u>	144
<u>5.9.1 Les classes : class</u>	144
<u>5.9.2 extends : héritage</u>	148
<u>5.9.3 Constructeur</u>	148
<u>5.9.4 Opérateur de contexte de classe (::)</u>	150
<u>5.9.5 parent</u>	150
<u>5.9.6 Sauvegarde d'objets - cas des sessions</u>	151
<u>5.9.7 Les fonctions magiques sleep et wakeup</u>	152
<u>5.9.8 Références dans un constructeur</u>	152
<u>5.9.9 Comparer des objets</u>	155
<u>5.10 Les classes et les objets (PHP 5)</u>	157
<u>5.10.1 Introduction</u>	157
<u>5.10.2 Syntaxe de base</u>	157
<u>5.10.3 Auto-chargement de classes</u>	160
<u>5.10.4 Constructeurs et destructeurs</u>	160
<u>5.10.5 Visibilité</u>	162
<u>5.10.6 L'opérateur de résolution de portée (::)</u>	164
<u>5.10.7 Statique</u>	165
<u>5.10.8 Constantes de classe</u>	166
<u>5.10.9 Abstraction de classes</u>	167
<u>5.10.10 Interfaces</u>	168
<u>5.10.11 Surcharge</u>	169
<u>5.10.12 Parcours d'objets</u>	172
<u>5.10.13 Masques</u>	175
<u>5.10.14 Méthodes magiques</u>	177
<u>5.10.15 Mot clé "final"</u>	179
<u>5.10.16 Clonage d'objets</u>	179
<u>5.10.17 Comparaison d'objets</u>	181

Sommaire

5 Référence du langage	
5.10.18 Réflexion	183
5.10.19 Typage objet	194
5.11 Exceptions	195
5.11.1 Exceptions étendues	196
5.12 Les références	198
5.12.1 Qu'est ce qu'une référence?	198
5.12.2 Que font les références ?	198
5.12.3 Ce que les références ne sont pas	200
5.12.4 Passage par référence	201
5.12.5 Retourner des références	202
5.12.6 Détruire une référence	202
5.12.7 Repérer une référence	203
6 Sécurité	204
6.1 Introduction	204
6.2 Considérations générales	204
6.3 Binaires CGI	205
6.3.1 Faiblesses connues	205
6.3.2 Cas 1 : Tous les fichiers sont publics	206
6.3.3 Cas 2 : Utilisation de la directive de compilation <code>--enable-force-cgi-redirect</code>	206
6.3.4 Cas 3 : Utilisation du "doc_root" ou du "user_dir"	206
6.3.5 Cas 4 : L'exécutable PHP à l'extérieur de l'arborescence du serveur	207
6.4 Installé en tant que module Apache	207
6.5 Sécurité des fichiers	208
6.6 Sécurité des bases de données	209
6.6.1 Schéma de base de données	210
6.6.2 Connexions au serveur de base de données	210
6.6.3 Modèle de stockage avec chiffrement	210
6.6.4 Injection SQL	211
6.7 Rapport d'erreurs	214
6.8 Utilisation des variables super-globales	216
6.9 Données transmises par les internautes	217
6.10 Guillemets magiques	218
6.10.1 Qu'est-ce que les guillemets magiques?	218
6.10.2 Pourquoi utiliser les guillemets magiques?	219
6.10.3 Pourquoi ne pas utiliser les guillemets magiques?	219
6.10.4 Désactiver les guillemets magiques	220
6.11 Masquer PHP	220
6.12 Etre à jour	221
7 Caractéristiques	222
7.1 Identification HTTP avec PHP	222
7.2 Cookies	225
7.3 Sessions	225
7.4 Utiliser les XForms	225
7.5 Gestion des chargements de fichier	226
7.5.1 Chargements de fichiers par méthode POST	226
7.5.2 Explication sur les messages d'erreurs de chargement de fichiers	229
7.5.3 Erreurs classiques	229
7.5.4 Télécharger plusieurs fichiers simultanément	230
7.5.5 Chargement par méthode PUT	231

Sommaire

7 Caractéristiques

<u>7.6 Utilisation des fichiers à distance</u>	232
<u>7.7 Gestion des connexions</u>	233
<u>7.8 Connexions persistantes aux bases de données</u>	234
<u>7.9 Safe mode</u>	235
<u>7.9.1 Sécurité et Safe Mode</u>	235
<u>7.9.2 Fonctions désactivées par le Safe Mode</u>	239
<u>7.10 Utiliser PHP en ligne de commande</u>	242

8 Référence des fonctions.....253

<u>8.1 Fonctions Apache</u>	253
<u>8.1.1 Introduction</u>	253
<u>8.1.2 Installation</u>	253
<u>8.1.3 Configuration à l'exécution</u>	253
<u>8.1.4 Types de ressources</u>	254
<u>8.1.5 Constantes pré-définies</u>	254
<u>8.1.7 apache_get_modules() : Retourne la liste des modules Apache chargés</u>	254
<u>8.1.8 apache_get_version() : Récupère la version d'Apache</u>	254
<u>8.1.9 apache_getenv() : Obtient une variable subprocess_env d'Apache</u>	255
<u>8.1.10 apache_lookup_uri() : Effectue une requête partielle pour l'URI spécifiée et renvoie toutes les informations</u>	255
<u>8.1.11 apache_note() : Affiche ou affecte le paramètre "apache request notes"</u>	255
<u>8.1.12 apache_request_headers() : Récupère tous les en-têtes HTTP de la requête</u>	255
<u>8.1.13 apache_reset_timeout() : Remet à sa position initiale le temporisateur d'Apache</u>	255
<u>8.1.14 apache_response_headers() : Récupère tous les en-têtes de réponse HTTP</u>	256
<u>8.1.15 apache_setenv() : Modifie une variable de subprocess_env Apache</u>	256
<u>8.1.16 ascii2ebcdic() : Transforme une chaîne ASCII en EBCDIC</u>	256
<u>8.1.17 ebcdic2ascii() : Transforme une chaîne EBCDIC en ASCII</u>	256
<u>8.1.18 getallheaders() : Récupère tous les en-têtes de la requête HTTP</u>	256
<u>8.1.19 virtual() : Effectue une sous-requête Apache</u>	257
<u>8.2 Débugueur avancé pour PHP</u>	258
<u>8.2.1 Introduction</u>	258
<u>8.2.2 Installation</u>	258
<u>8.2.3 Compilation sous un environnement Win32</u>	258
<u>8.2.4 Configuration à l'exécution</u>	259
<u>8.2.5 Types de ressources</u>	259
<u>8.2.6 Constantes pré-définies</u>	259
<u>8.2.7 Comment utiliser APD dans vos scripts ?</u>	259
<u>8.2.8 Informations de contact</u>	260
<u>8.2.10 apd_callstack() : Retourne la pile d'appel courante dans un tableau</u>	261
<u>8.2.11 apd_clunk() : Lance une alerte et un callstack</u>	261
<u>8.2.12 apd_continue() : Redémarre l'interpréteur</u>	261
<u>8.2.13 apd_croak() : Lance une erreur, un callstack et sort</u>	261
<u>8.2.14 apd_dump_function_table() : Affiche la table courant de fonction</u>	262
<u>8.2.15 apd_dump_persistent_resources() : Retourne toutes les ressources persistantes dans un tableau</u>	262
<u>8.2.16 apd_dump_regular_resources() : Retourne toutes les ressources régulières courantes dans un tableau</u>	262
<u>8.2.17 apd_echo() : Ecrit dans le socket de débogage</u>	262

Sommaire

8 Référence des fonctions

<u>8.2.18 apd_get_active_symbols()</u> : Récupère un tableau contenant les noms de variables courantes de portées locales.....	262
<u>8.2.19 apd_set_pprof_trace()</u> : Démarre la session de débogage APD.....	263
<u>8.2.20 apd_set_session_trace()</u> : Démarre la session de débogage.....	263
<u>8.2.21 apd_set_session()</u> : Modifie ou définit le degré de débogage courant.....	264
<u>8.2.22 apd_set_socket_session_trace()</u> : Démarre la session de débogage à distance.....	264
<u>8.2.23 override_function()</u> : Dépasse les fonctions intégrées.....	265
<u>8.2.24 rename_function()</u> : Renomme une fonction intégrée dans la table des fonctions globales.....	265
8.3 Tableaux.....	266
<u>8.3.1 Introduction.....</u>	<u>266</u>
<u>8.3.2 Pré-requis.....</u>	<u>266</u>
<u>8.3.3 Installation.....</u>	<u>266</u>
<u>8.3.4 Configuration à l'exécution.....</u>	<u>266</u>
<u>8.3.5 Types de ressources.....</u>	<u>266</u>
<u>8.3.6 Constantes pré-définies.....</u>	<u>266</u>
<u>8.3.7 Voir aussi.....</u>	<u>267</u>
<u>8.3.9 array_chunk()</u> : Sépare un tableau en tableaux de taille inférieure.....	269
<u>8.3.10 array_combine()</u> : Crée un tableau à partir de deux autres tableaux.....	270
<u>8.3.11 array_count_values()</u> : Compte le nombre de valeurs dans un tableau.....	270
<u>8.3.12 array_diff_assoc()</u> : Calcule la différence de deux tableaux, en prenant en compte les clés.....	271
<u>8.3.13 array_diff_key()</u> : Calcul la différence de deux tableaux en utilisant les clés pour comparaison.....	272
<u>8.3.14 array_diff_uassoc()</u> : Calcule la différence entre deux tableaux associatifs, à l'aide d'une fonction utilisateur.....	272
<u>8.3.15 array_diff_ukey()</u> : Calcule la différence entre deux tableaux en utilisant une fonction de callback sur les clés pour comparaison.....	273
<u>8.3.16 array_diff()</u> : Calcule la différence entre deux tableaux.....	274
<u>8.3.17 array_fill()</u> : Remplis un tableau avec une même valeur.....	275
<u>8.3.18 array_filter()</u> : Filtre les éléments d'un tableau.....	276
<u>8.3.19 array_flip()</u> : Remplace les clés par les valeurs, et les valeurs par les clés.....	277
<u>8.3.20 array_intersect_assoc()</u> : Calcule l'intersection de deux tableaux avec des tests sur les index.....	278
<u>8.3.21 array_intersect_key()</u> : Calcul l'intersection de deux tableaux en utilisant les clés pour comparaison.....	278
<u>8.3.22 array_intersect_uassoc()</u> : Calcul l'intersection de deux tableaux avec des tests sur les index, compare les index en utilisant une fonction de callback.....	279
<u>8.3.23 array_intersect_ukey()</u> : Calcul l'intersection de deux tableaux en utilisant une fonction de callback sur les clés pour comparaison.....	280
<u>8.3.24 array_intersect()</u> : Calcule l'intersection de tableaux.....	281
<u>8.3.25 array_key_exists()</u> : Vérifie si une clé existe dans un tableau.....	281
<u>8.3.26 array_keys()</u> : Retourne toutes les clés d'un tableau.....	282
<u>8.3.27 array_map()</u> : Applique une fonction sur les éléments d'un tableau.....	283
<u>8.3.28 array_merge_recursive()</u> : Combine plusieurs tableaux ensemble, récursivement.....	285
<u>8.3.29 array_merge()</u> : Fusionne un ou plusieurs tableaux.....	286
<u>8.3.30 array_multisort()</u> : Tri multi-dimensionnel de tableaux.....	288
<u>8.3.31 array_pad()</u> : Complète un tableau avec une valeur jusqu'à la longueur spécifiée.....	291

Sommaire

8 Référence des fonctions

<u>8.3.32 array_pop()</u> : Dépile un élément de la fin d'un tableau.....	292
<u>8.3.33 array_product()</u> : Calcule le produit des valeurs du tableau.....	292
<u>8.3.34 array_push()</u> : Empile un ou plusieurs éléments à la fin d'un tableau.....	293
<u>8.3.35 array_rand()</u> : Prend une ou plusieurs valeurs, au hasard dans un tableau.....	293
<u>8.3.36 array_reduce()</u> : Réduit itérativement un tableau.....	294
<u>8.3.37 array_reverse()</u> : Inverse l'ordre des éléments d'un tableau.....	295
<u>8.3.38 array_search()</u> : Recherche dans un tableau la clé associée à une valeur.....	295
<u>8.3.39 array_shift()</u> : Dépile un élément au début d'un tableau.....	296
<u>8.3.40 array_slice()</u> : Extrait une portion de tableau.....	297
<u>8.3.41 array_splice()</u> : Efface et remplace une portion de tableau.....	298
<u>8.3.42 array_sum()</u> : Calcule la somme des valeurs du tableau.....	299
<u>8.3.43 array_udiff_assoc()</u> : Calcule la différence entre des tableaux avec vérification des index, compare les données avec une fonction de callback.....	299
<u>8.3.44 array_udiff_uassoc()</u> : Calcule la différence de deux tableaux associatifs, compare les données et les index avec une fonction de callback.....	300
<u>8.3.45 array_udiff()</u> : Calcule la différence entre deux tableaux en utilisant une fonction callback.....	302
<u>8.3.46 array_uintersect_assoc()</u> : Calcul l'intersection de deux tableaux avec des tests sur l'index, compare les donnée en utilisant une fonction de callback.....	303
<u>8.3.47 array_uintersect_uassoc()</u> : Calcul l'intersection de deux tableaux avec des tests sur l'index, compare les données et les indexes des deux tableaux en utilisant une fonction de callback.....	303
<u>8.3.48 array_uintersect()</u> : Calcul l'intersection de deux tableaux, compare les données en utilisant une fonction de callback.....	304
<u>8.3.49 array_unique()</u> : Dédoublonne un tableau.....	305
<u>8.3.50 array_unshift()</u> : Empile un ou plusieurs éléments au début d'un tableau.....	306
<u>8.3.51 array_values()</u> : Retourne les valeurs d'un tableau.....	306
<u>8.3.52 array_walk_recursive()</u> : Applique une fonction utilisateur récursivement à chaque membre du tableau.....	307
<u>8.3.53 array_walk()</u> : Exécute une fonction sur chacun des éléments d'un tableau.....	307
<u>8.3.54 array()</u> : Crée un tableau.....	309
<u>8.3.55 arsort()</u> : Trie un tableau en ordre inverse.....	310
<u>8.3.56 asort()</u> : Trie un tableau et conserve l'association des index.....	311
<u>8.3.57 compact()</u> : Crée un tableau à partir de variables et de leur valeur.....	312
<u>8.3.58 count()</u> : Compte le nombre d'éléments d'un tableau ou le nombre de propriétés d'un objet.....	312
<u>8.3.59 current()</u> : Retourne l'élément courant du tableau.....	314
<u>8.3.60 each()</u> : Retourne chaque paire clé/valeur d'un tableau.....	314
<u>8.3.61 end()</u> : Positionne le pointeur de tableau en fin de tableau.....	316
<u>8.3.62 extract()</u> : Importe les variables dans la table des symboles.....	316
<u>8.3.63 in_array()</u> : Indique si une valeur appartient à un tableau.....	318
<u>8.3.64 key()</u> : Retourne une clé d'un tableau associatif.....	319
<u>8.3.65 krsort()</u> : Trie un tableau en sens inverse et suivant les clés.....	319
<u>8.3.66 ksort()</u> : Trie un tableau suivant les clés.....	320
<u>8.3.67 list()</u> : Transforme une liste de variables en tableau.....	320
<u>8.3.68 natcasesort()</u> : Trie un tableau avec l'algorithme à "ordre naturel" insensible à la casse.....	322
<u>8.3.69 natsort()</u> : Trie un tableau avec l'algorithme à "ordre naturel".....	323
<u>8.3.70 next()</u> : Avance le pointeur interne d'un tableau.....	324
<u>8.3.71 pos()</u> : Alias de current.....	324
<u>8.3.72 prev()</u> : Recule le pointeur courant de tableau.....	324

Sommaire

8 Référence des fonctions

<u>8.3.73 range()</u> : Crée un tableau contenant un intervalle d'éléments.....	325
<u>8.3.74 reset()</u> : Remet le pointeur interne de tableau au début.....	326
<u>8.3.75 rsort()</u> : Trie un tableau en ordre inverse.....	326
<u>8.3.76 shuffle()</u> : Mélange les éléments d'un tableau.....	327
<u>8.3.77 sizeof()</u> : Alias de count.....	327
<u>8.3.78 sort()</u> : Trie un tableau.....	327
<u>8.3.79 uasort()</u> : Trie un tableau en utilisant une fonction de callback.....	328
<u>8.3.80 uksort()</u> : Trie un tableau par ses clés en utilisant une fonction de callback.....	329
<u>8.3.81 usort()</u> : Trie un tableau en utilisant une fonction de comparaison.....	330
<u>8.4 Aspell [Obs.]</u>	333
<u>8.4.1 Introduction</u>	333
<u>8.4.2 Pré-requis</u>	333
<u>8.4.3 Installation</u>	333
<u>8.4.4 Voir aussi</u>	333
<u>8.4.6 aspell_check()</u> : Vérifie un mot [Obsolète].....	333
<u>8.4.7 aspell_new()</u> : Charge un nouveau dictionnaire [Obsolète].....	334
<u>8.4.8 aspell_suggest()</u> : Suggère l'orthographe d'un mot [Obsolète].....	334
<u>8.5 Nombres de grande taille BCMath</u>	335
<u>8.5.1 Introduction</u>	335
<u>8.5.2 Pré-requis</u>	335
<u>8.5.3 Installation</u>	335
<u>8.5.4 Configuration à l'exécution</u>	335
<u>8.5.5 Types de ressources</u>	335
<u>8.5.6 Constantes pré-définies</u>	335
<u>8.5.8 bccomp()</u> : Compare deux nombres de grande taille.....	336
<u>8.5.9 bcdiv()</u> : Divise deux nombres de grande taille.....	336
<u>8.5.10 bcmathmod()</u> : Retourne le reste d'une division entre nombres de grande taille.....	336
<u>8.5.11 bcmul()</u> : Multiplie deux nombres de grande taille.....	336
<u>8.5.12 bcpow()</u> : Elève un nombre à la puissance n-ième.....	337
<u>8.5.13 bcpowmod()</u> : Calcule le reste modulo d'un nombre élevé à une puissance.....	337
<u>8.5.14 bcscale()</u> : Détermine le nombre de décimales par défaut.....	337
<u>8.5.15 bcsqrt()</u> : Renvoie la racine carrée d'un nombre de grande taille.....	337
<u>8.5.16 bcsub()</u> : Soustrait un nombre de grande taille à un autre.....	337
<u>8.6 Compilateur bytecode pour PHP</u>	338
<u>8.6.1 Introduction</u>	338
<u>8.6.2 Installation</u>	338
<u>8.6.3 Contact</u>	339
<u>8.6.5 bcompiler_load()</u> : Lit et crée les classes depuis un fichier compressé bzippé.....	339
<u>8.6.6 bcompiler_parse_class()</u> : Lit le bytecode d'une classe et revient à une fonction utilisateur.....	340
<u>8.6.7 bcompiler_read()</u> : Lit et crée les classes depuis un descripteur de fichier.....	340
<u>8.6.8 bcompiler_write_class()</u> : Ecrit une classe définie en bytecode.....	341
<u>8.6.9 bcompiler_write_constant()</u> : Ecrit une constante définie comme bytecode.....	341
<u>8.6.10 bcompiler_write_exe_footer()</u> : Ecrit à la position start et jusqu'à la fin du fichier de type exe.....	342
<u>8.6.11 bcompiler_write_file()</u> : Ecrit un code source PHP sous forme de bytecodes.....	343
<u>8.6.12 bcompiler_write_footer()</u> : Ecrit le caractère \x00 pour indiquer la fin des données compilées.....	343
<u>8.6.13 bcompiler_write_function()</u> : Ecrit une fonction définie sous forme de bytecode.....	344
<u>8.6.14 bcompiler_write_functions_from_file()</u> : Ecrit toutes les fonctions définies	

Sommaire

8 Référence des fonctions

<u> dans un fichier sous forme de bytecode</u>	344
<u> 8.6.15 bcompiler_write_header() : Ecrit l'en-tête bcompiler</u>	345
8.7 Compression Bzip2	346
<u>8.7.1 Introduction</u>	346
<u>8.7.2 Pré-requis</u>	346
<u>8.7.3 Installation</u>	346
<u>8.7.4 Configuration à l'exécution</u>	346
<u>8.7.5 Types de ressources</u>	346
<u>8.7.6 Constantes pré-définies</u>	346
<u>8.7.7 Exemples</u>	346
<u>8.7.9 bzcompress() : Comprime une chaîne avec bzip2</u>	347
<u>8.7.10 bzdecompress() : Décompressé une chaîne bzip2</u>	347
<u>8.7.11 bzerrno() : Retourne le code d'erreur bzip2</u>	347
<u>8.7.12 bzerror() : Retourne le numéro et le message d'erreur bzip2 dans un tableau</u>	348
<u>8.7.13 bzerrstr() : Retourne le message d'erreur bzip2</u>	348
<u>8.7.14 bzflush() : Force l'écriture de toutes les données compressées</u>	348
<u>8.7.15 bzopen() : Ouvre un fichier compressé avec bzip2</u>	348
<u>8.7.16 bzread() : Lecture binaire d'un fichier bzip2</u>	348
<u>8.7.17 bzwrite() : Ecriture binaire dans un fichier bzip2</u>	348
8.8 Calendrier	349
<u>8.8.1 Introduction</u>	349
<u>8.8.2 Installation</u>	349
<u>8.8.3 Configuration à l'exécution</u>	349
<u>8.8.4 Types de ressources</u>	349
<u>8.8.5 Constantes pré-définies</u>	349
<u>8.8.7 cal_from_jd() : Convertit le nombre de jours Julien en un calendrier et retourne des détails</u>	350
<u>8.8.8 cal_info() : Retourne des détails sur un calendrier</u>	351
<u>8.8.9 cal_to_jd() : Convertit un calendrier en nombre de jours Julien</u>	352
<u>8.8.10 easter_date() : Retourne un timestamp UNIX pour Pâques, à minuit</u>	352
<u>8.8.11 easter_days() : Retourne le nombre de jours entre le 21 Mars et Pâques, pour une année donnée</u>	353
<u>8.8.12 FrenchToJD() : Convertit une date du calendrier français républicain en nombre de jours du calendrier Julien</u>	354
<u>8.8.13 GregorianToJD() : Convertit une date grégorienne en nombre de jours du calendrier Julien</u>	354
<u>8.8.14 JDDayOfWeek() : Retourne le numéro du jour de la semaine</u>	354
<u>8.8.15 JDMonthName() : Retourne le nom du mois</u>	355
<u>8.8.16 JDToFrench() : Convertit le nombre de jours du calendrier Julien en date du calendrier français républicain</u>	355
<u>8.8.17 JDToGregorian() : Convertit le nombre de jours du calendrier Julien en date grégorienne</u>	355
<u>8.8.18 jdtojewish() : Convertit le nombre de jours du calendrier Julien en date du calendrier juif</u>	355
<u>8.8.19 JDToJulian() : Convertit le nombre de jours du calendrier Julien en date du calendrier Julien</u>	356
<u>8.8.20 jdtounix() : Convertit un jours Julien en timestamp UNIX</u>	356
<u>8.8.21 JewishToJD() : Convertit une date du calendrier Juif en nombre de jours du calendrier Julien</u>	356
<u>8.8.22 JulianToJD() : Convertit un jours du calendrier Julien en un nombre de</u>	

Sommaire

8 Référence des fonctions

<u>jours du calendrier Julien</u>	356
<u>8.8.23 unixtojd() : Convertit un timestamp UNIX en un jours Julien</u>	357
8.9 Paiement CCVS [déconseillé]	358
<u>8.9.1 Introduction</u>	358
<u>8.9.2 Installation</u>	358
<u>8.9.3 Voir aussi</u>	358
<u>8.9.5 ccvs_auth() : Effectue un test d'autorisation de crédit sur une transaction</u>	359
<u>8.9.6 ccvs_command() : Exécute une commande spécifique à un protocole et qui n'est donc pas disponible dans l'API générique</u>	359
<u>8.9.7 ccvs_count() : Trouve le nombre de transactions d'un certain type stockées dans le système</u>	359
<u>8.9.8 ccvs_delete() : Supprime une transaction</u>	359
<u>8.9.9 ccvs_done() : Termine l'enfin CCVS et libère les ressources</u>	359
<u>8.9.10 ccvs_init() : Initialise CCVS pour utilisation</u>	360
<u>8.9.11 ccvs_lookup() : Recherche un élément d'un type spécifique dans la base de données</u>	360
<u>8.9.12 ccvs_new() : Crée une nouvelle transaction à vide</u>	360
<u>8.9.13 ccvs_report() : Retourne le status du processus de communication</u>	360
<u>8.9.14 ccvs_return() : Transfère des fonds de chez le marchand vers le possesseur de la carte de crédit</u>	360
<u>8.9.15 ccvs_reverse() : Annule entièrement une autorisation déjà donnée</u>	360
<u>8.9.16 ccvs_sale() : Transfère des fonds de chez le possesseur de la carte de crédit vers le marchand</u>	361
<u>8.9.17 ccvs_status() : Vérifie le status d'une facture</u>	361
<u>8.9.18 ccvs_textvalue() : Retourne la valeur textuelle du dernier appel à une fonction</u>	361
<u>8.9.19 ccvs_void() : Annule complètement une transaction complétée</u>	361
8.10 Fonctions Classkit	362
<u>8.10.1 Introduction</u>	362
<u>8.10.2 Installation</u>	362
<u>8.10.3 Configuration à l'exécution</u>	362
<u>8.10.4 Types de ressources</u>	362
<u>8.10.5 Constantes pré-définies</u>	362
<u>8.10.7 classkit_method_add() : Ajoute dynamiquement une nouvelle méthode à une classe donnée</u>	363
<u>8.10.8 classkit_method_copy() : Copie une méthode d'une classe vers une autre classe</u>	363
<u>8.10.9 classkit_method_redefine() : Change dynamiquement le code de la méthode donnée</u>	363
<u>8.10.10 classkit_method_remove() : Efface dynamiquement une méthode donnée</u>	363
<u>8.10.11 classkit_method_rename() : Change dynamiquement le nom d'une méthode donnée</u>	364
8.11 Classes et Objets	365
<u>8.11.1 Introduction</u>	365
<u>8.11.2 Pré-requis</u>	365
<u>8.11.3 Installation</u>	365
<u>8.11.4 Configuration à l'exécution</u>	365
<u>8.11.5 Types de ressources</u>	365
<u>8.11.6 Constantes pré-définies</u>	365
<u>8.11.7 Exemples</u>	365
<u>8.11.9 call_user_method() : Appelle une méthode utilisateur d'un objet</u>	368

Sommaire

8 Référence des fonctions

<u>8.11.10 class_exists()</u> : Vérifie qu'une classe a été définie.....	368
<u>8.11.11 get_class_methods()</u> : Retourne les noms des méthodes d'une classe.....	369
<u>8.11.12 get_class_vars()</u> : Retourne les valeurs par défaut des propriétés d'une classe.....	370
<u>8.11.13 get_class()</u> : Retourne la classe d'un objet.....	371
<u>8.11.14 get_declared_classes()</u> : Liste toutes les classes définies.....	372
<u>8.11.15 get_declared_interfaces()</u> : Retourne un tableau avec toutes les interfaces déclarées.....	372
<u>8.11.16 get_object_vars()</u> : Retourne un tableau associatif des propriétés d'un objet.....	373
<u>8.11.17 get_parent_class()</u> : Retourne le nom de la classe d'un objet.....	374
<u>8.11.18 interface_exists()</u> : Vérifie si une interface a été définie.....	375
<u>8.11.19 is_a()</u> : Retourne TRUE si un objet a pour parent une classe donnée.....	375
<u>8.11.20 is_subclass_of()</u> : Détermine si un objet est une sous-classe.....	376
<u>8.11.21 method_exists()</u> : Vérifie que la méthode existe pour une classe.....	377
<u>8.11.22 property_exists()</u> : Vérifie si un objet ou une classe a une propriété.....	377
<u>8.12 COM et .Net (Windows)</u>	378
<u>8.12.1 Introduction</u>	378
<u>8.12.2 Pré-requis</u>	378
<u>8.12.3 Installation</u>	378
<u>8.12.4 For Each</u>	379
<u>8.12.5 Tableaux et propriétés à la façon des tableaux de COM</u>	379
<u>8.12.6 Exceptions (PHP 5)</u>	379
<u>8.12.7 Configuration à l'exécution</u>	380
<u>8.12.8 Constantes pré-définies</u>	381
<u>8.12.9 Voir aussi</u>	382
<u>8.12.11 DOTNET()</u> : La classe DOTNET.....	383
<u>8.12.12 VARIANT()</u> : classe VARIANT.....	383
<u>8.12.13 com_addrf()</u> : Incrémente le compteur de références (déconseillé).....	383
<u>8.12.14 com_create_guid()</u> : Génère un identifiant unique global.....	383
<u>8.12.15 com_event_sink()</u> : Connecte des événements d'un objet COM sur un objet PHP.....	384
<u>8.12.16 com_get_active_object()</u> : Retourne un objet représentant l'instance actuelle d'un objet COM.....	385
<u>8.12.17 com_get()</u> : Lit la valeur d'une propriété d'un composant COM.....	385
<u>8.12.18 com_invoke()</u> : Appelle une méthode d'un composant (déconseillé).....	385
<u>8.12.19 com_isenum()</u> : Récupère un IEnumVariant (déconseillé).....	386
<u>8.12.20 com_load_typelib()</u> : Charge un Typelib.....	386
<u>8.12.21 com_load()</u> : Crée une référence sur un composant COM (déconseillé).....	387
<u>8.12.22 com_message_pump()</u> : Traite un message COM en attendant jusqu'à timeoutms millisecondes.....	387
<u>8.12.23 com_print_typeinfo()</u> : Affiche une définition de classe PHP pour une interface répartissable.....	387
<u>8.12.24 com_propget()</u> : Alias de com_get.....	388
<u>8.12.25 com_propput()</u> : Alias de com_set.....	388
<u>8.12.26 com_propset()</u> : Alias de com_set.....	388
<u>8.12.27 com_release()</u> : Décrémente le compteur de références.....	388
<u>8.12.28 com_set()</u> : Modifie une propriété d'un composant COM.....	388
<u>8.12.29 variant_abs()</u> : Retourne la valeur absolue d'un variant.....	389
<u>8.12.30 variant_add()</u> : "Ajoute" deux valeurs de variants et retourne le résultat.....	389
<u>8.12.31 variant_and()</u> : Effectue un ET entre deux variants et retourne le résultat.....	390

Sommaire

8 Référence des fonctions

<u>8.12.32 variant_cast()</u> : Convertit un variant en un nouvel objet variant de type différent.....	390
<u>8.12.33 variant_cat()</u> : Concatène deux valeurs variantes ensemble et retourne le résultat.....	391
<u>8.12.34 variant_cmp()</u> : Compare deux variants.....	391
<u>8.12.35 variant_date_from_timestamp()</u> : Retourne une représentation date en variant d'un timestamp unix.....	392
<u>8.12.36 variant_date_to_timestamp()</u> : Convertit une valeur date/temps variante en un timestamp unix.....	392
<u>8.12.37 variant_div()</u> : Retourne le résultat de la division de deux variants.....	392
<u>8.12.38 variant_eqv()</u> : Performe une équivalence de bits de deux variants.....	393
<u>8.12.39 variant_fix()</u> : Retourne la portion entière d'un variant.....	393
<u>8.12.40 variant_get_type()</u> : Retourne le type d'un objet variant.....	394
<u>8.12.41 variant_idiv()</u> : Convertit des variants en valeurs entières, et performe alors une division.....	394
<u>8.12.42 variant_imp()</u> : Performe une implication sur les bits de deux variants.....	395
<u>8.12.43 variant_int()</u> : Retourne la partie entière d'un variant.....	395
<u>8.12.44 variant_mod()</u> : Divise deux variantes et retourne le reste.....	396
<u>8.12.45 variant_mul()</u> : Multiplie les valeurs de deux variants et retourne le résultat.....	396
<u>8.12.46 variant_neg()</u> : Effectue une négation logique sur un variant.....	397
<u>8.12.47 variant_not()</u> : Performe une négation sur les bits sur un variant.....	397
<u>8.12.48 variant_or()</u> : Performe une disjonction logique sur deux variants.....	398
<u>8.12.49 variant_pow()</u> : Retourne le résultat de la fonction puissance avec deux variants.....	398
<u>8.12.50 variant_round()</u> : Arrondit le variant au nombre spécifié de décimales.....	399
<u>8.12.51 variant_set_type()</u> : Convertit un variant en un autre type "sur-place".....	399
<u>8.12.52 variant_set()</u> : Assigne une nouvelle valeur pour un objet variant.....	399
<u>8.12.53 variant_sub()</u> : Soustrait la valeur du variant de droite de la valeur de celui de gauche et retourne le résultat.....	399
<u>8.12.54 variant_xor()</u> : Performe une exclusion logique sur deux variants.....	400
<u>8.13 Fonctions ClibPDF</u>	401
<u>8.13.1 Introduction</u>	401
<u>8.13.2 Pré-requis</u>	401
<u>8.13.3 Installation</u>	401
<u>8.13.4 Configuration à l'exécution</u>	402
<u>8.13.5 Constantes pré-définies</u>	402
<u>8.13.6 Exemples</u>	402
<u>8.13.7 Voir aussi</u>	404
<u>8.13.9 cpdf_add_outline()</u> : Ajoute un signet à la page courante.....	405
<u>8.13.10 cpdf_arc()</u> : Dessine un arc de cercle.....	406
<u>8.13.11 cpdf_begin_text()</u> : Démarre une section de texte.....	406
<u>8.13.12 cpdf_circle()</u> : Dessine un cercle.....	406
<u>8.13.13 cpdf_clip()</u> : Aligne les dessins sur le chemin courant.....	407
<u>8.13.14 cpdf_close()</u> : Ferme un fichier PDF.....	407
<u>8.13.15 cpdf_closepath_fill_stroke()</u> : Remplit le chemin, dessine le bord et ferme le chemin.....	407
<u>8.13.16 cpdf_closepath_stroke()</u> : Ferme le fichier et dessine une ligne le long du chemin.....	407
<u>8.13.17 cpdf_closepath()</u> : Ferme le chemin.....	407
<u>8.13.18 cpdf_continue_text()</u> : Affiche le texte à la ligne suivante.....	408

Sommaire

8 Référence des fonctions

<u>8.13.19 cpdf_curveto()</u> : Dessine une courbe.....	408
<u>8.13.20 cpdf_end_text()</u> : Termine une section de texte.....	408
<u>8.13.21 cpdf_fill_stroke()</u> : Remplit le chemin, et dessine le bord.....	408
<u>8.13.22 cpdf_fill()</u> : Remplit le chemin courant.....	409
<u>8.13.23 cpdf_finalize_page()</u> : Termine une page.....	409
<u>8.13.24 cpdf_finalize()</u> : Termine un document.....	409
<u>8.13.25 cpdf_global_set_document_limits()</u> : Fixe les limites d'un document PDF.....	409
<u>8.13.26 cpdf_import_jpeg()</u> : Ouvre une image JPEG.....	409
<u>8.13.27 cpdf_lineto()</u> : Dessine une ligne.....	410
<u>8.13.28 cpdf_moveto()</u> : Fixe le point courant de dessin CPDF.....	410
<u>8.13.29 cpdf_newpath()</u> : Commence un nouveau chemin.....	410
<u>8.13.30 cpdf_open()</u> : Ouvre un nouveau document PDF.....	410
<u>8.13.31 cpdf_output_buffer()</u> : Affiche le document PDF qui est en mémoire.....	411
<u>8.13.32 cpdf_page_init()</u> : Commence une nouvelle page.....	411
<u>8.13.33 cpdf_place_inline_image()</u> : Place une image dans une page.....	411
<u>8.13.34 cpdf_rect()</u> : Dessine un rectangle.....	412
<u>8.13.35 cpdf_restore()</u> : Restaure un environnement sauvegardé.....	412
<u>8.13.36 cpdf_rlineto()</u> : Dessine une ligne, relativement.....	413
<u>8.13.37 cpdf_rmoveto()</u> : Fixe le point courant relativement.....	413
<u>8.13.38 cpdf_rotate_text()</u> : Configure l'inclinaison d'un texte.....	413
<u>8.13.39 cpdf_rotate()</u> : Effectue une rotation.....	413
<u>8.13.40 cpdf_save_to_file()</u> : Ecrit un document PDF dans un fichier.....	414
<u>8.13.41 cpdf_save()</u> : Sauve l'environnement courant.....	414
<u>8.13.42 cpdf_scale()</u> : Modifie l'échelle.....	414
<u>8.13.43 cpdf_set_action_url()</u> : Paramètre un hyper lien.....	414
<u>8.13.44 cpdf_set_char_spacing()</u> : Fixe l'espacement des caractères.....	414
<u>8.13.45 cpdf_set_creator()</u> : Fixe le créateur d'un document PDF.....	414
<u>8.13.46 cpdf_set_current_page()</u> : Fixe la page courante.....	415
<u>8.13.47 cpdf_set_font_directories()</u> : Précise les dossiers à fouiller pour rechercher des polices externes.....	415
<u>8.13.48 cpdf_set_font_map_file()</u> : Configure le fichier de remplacement de polices externes.....	415
<u>8.13.49 cpdf_set_font()</u> : Sélectionne la police courante et sa taille.....	415
<u>8.13.50 cpdf_set_horiz_scaling()</u> : Fixe l'échelle horizontale du texte.....	416
<u>8.13.51 cpdf_set_keywords()</u> : Fixe les mots-clés d'un document PDF.....	416
<u>8.13.52 cpdf_set_leading()</u> : Fixe la distance entre deux lignes.....	416
<u>8.13.53 cpdf_set_page_animation()</u> : Fixe l'animation de la transition entre les pages.....	416
<u>8.13.54 cpdf_set_subject()</u> : Fixe le sujet d'un document PDF.....	417
<u>8.13.55 cpdf_set_text_matrix()</u> : Fixe la matrice du texte.....	417
<u>8.13.56 cpdf_set_text_pos()</u> : Fixe la position du texte.....	417
<u>8.13.57 cpdf_set_text_rendering()</u> : Détermine le rendu du texte.....	417
<u>8.13.58 cpdf_set_text_rise()</u> : Fixe l'élévation du texte.....	417
<u>8.13.59 cpdf_set_title()</u> : Fixe le titre d'un document PDF.....	418
<u>8.13.60 cpdf_set_viewer_preferences()</u> : Indique comment afficher le document à l'utilisateur.....	418
<u>8.13.61 cpdf_set_word_spacing()</u> : Fixe l'espacement des mots.....	418
<u>8.13.62 cpdf_setdash()</u> : Fixe le motif de pointillés.....	418
<u>8.13.63 cpdf_setflat()</u> : Fixe la platitude (flatness).....	418
<u>8.13.64 cpdf_setgray_fill()</u> : Modifie le niveau de gris comme couleur de remplissage.....	418

Sommaire

8 Référence des fonctions

8.13.65 <u>cpdf_setgray_stroke()</u> : Choisit un niveau de gris comme couleur de dessin.....	419
8.13.66 <u>cpdf_setgray()</u> : Modifie un niveau de gris comme couleur de dessin et de remplissage.....	419
8.13.67 <u>cpdf_setlinecap()</u> : Fixe le paramètre linecap.....	419
8.13.68 <u>cpdf_setlinejoin()</u> : Fixe le paramètre linejoin.....	419
8.13.69 <u>cpdf_setlinewidth()</u> : Fixe la largeur de ligne.....	419
8.13.70 <u>cpdf_setmiterlimit()</u> : Fixe le paramètre miter limit.....	420
8.13.71 <u>cpdf_setrgbcolor_fill()</u> : Choisit une couleur rgb comme couleur de remplissage.....	420
8.13.72 <u>cpdf_setrgbcolor_stroke()</u> : Choisit une couleur rgb comme couleur de dessin.....	420
8.13.73 <u>cpdf_setrgbcolor()</u> : Choisit une couleur rgb comme couleur de dessin et de remplissage.....	420
8.13.74 <u>cpdf_show_xy()</u> : Affiche un texte à une position.....	421
8.13.75 <u>cpdf_show()</u> : Imprime un texte à la position courante.....	421
8.13.76 <u>cpdf_stringwidth()</u> : Retourne la taille de la police courante.....	421
8.13.77 <u>cpdf_stroke()</u> : Dessine une ligne le long du chemin.....	421
8.13.78 <u>cpdf_text()</u> : Imprime un texte avec des options.....	422
8.13.79 <u>cpdf_translate()</u> : Modifie l'origine du système de coordonnées.....	422
8.14 <u>Crack</u>	423
8.14.1 <u>Introduction</u>	423
8.14.2 <u>Pré-requis</u>	423
8.14.3 <u>Installation</u>	423
8.14.4 <u>Configuration à l'exécution</u>	423
8.14.5 <u>Types de ressources</u>	423
8.14.6 <u>Constantes pré-définies</u>	424
8.14.7 <u>Exemples</u>	424
8.14.9 <u>crack_closedict()</u> : Referme le dictionnaire CrackLib.....	424
8.14.10 <u>crack_getlastmessage()</u> : Retourne le message de diagnostic.....	425
8.14.11 <u>crack_opendict()</u> : Ouvre un nouveau dictionnaire CrackLib.....	425
8.15 <u>Caractères</u>	426
8.15.1 <u>Introduction</u>	426
8.15.2 <u>Pré-requis</u>	426
8.15.3 <u>Installation</u>	426
8.15.4 <u>Configuration à l'exécution</u>	426
8.15.5 <u>Types de ressources</u>	427
8.15.6 <u>Constantes pré-définies</u>	427
8.15.8 <u>ctype_alpha()</u> : Vérifie qu'une chaîne est alphabétique.....	427
8.15.9 <u>ctype_cntrl()</u> : Vérifie qu'un caractère est un caractère de contrôle.....	427
8.15.10 <u>ctype_digit()</u> : Vérifie qu'une chaîne est un entier.....	427
8.15.11 <u>ctype_graph()</u> : Vérifie qu'une chaîne est imprimable.....	428
8.15.12 <u>ctype_lower()</u> : Vérifie qu'une chaîne est en minuscules.....	428
8.15.13 <u>ctype_print()</u> : Vérifie qu'une chaîne est imprimable.....	428
8.15.14 <u>ctype_punct()</u> : Vérifie qu'une chaîne contient de la ponctuation.....	428
8.15.15 <u>ctype_space()</u> : Vérifie qu'une chaîne n'est faite que de caractères blancs.....	428
8.15.16 <u>ctype_upper()</u> : Vérifie qu'une chaîne est en majuscules.....	428
8.15.17 <u>ctype_xdigit()</u> : Vérifie qu'un caractère représente un nombre hexadécimal.....	428
8.16 <u>CURL</u>	429
8.16.1 <u>Introduction</u>	429

Sommaire

8 Référence des fonctions

8.16.2	<u>Pré-requis</u>	429
8.16.3	<u>Installation</u>	429
8.16.4	<u>Constantes pré-définies</u>	429
8.16.5	<u>Exemples</u>	437
8.16.7	<u>curl_copy_handle() : Copie une ressource cURL avec toutes ses préférences</u>	438
8.16.8	<u>curl_errno() : Retourne un entier représentant le dernier message d'erreur CURL</u>	439
8.16.9	<u>curl_error() : Retourne une chaîne contenant le dernier message d'erreur CURL</u>	439
8.16.10	<u>curl_exec() : Exécute une session CURL</u>	439
8.16.11	<u>curl_getinfo() : Lit les informations détaillant un transfert CURL</u>	440
8.16.12	<u>curl_init() : Initialise une session CURL</u>	441
8.16.13	<u>curl_multi_add_handle() : Ajoute une ressource cURL à un cURL multiple</u>	441
8.16.14	<u>curl_multi_close() : Termine un jeu de sessions CURL</u>	441
8.16.15	<u>curl_multi_exec() : Exécute les sous-requêtes de la session CURL</u>	442
8.16.16	<u>curl_multi_getcontent() : Retourne le contenu obtenu avec l'option CURLOPT_RETURNTRANSFER</u>	442
8.16.17	<u>curl_multi_info_read() : Lit les informations sur les transferts actuels</u>	442
8.16.18	<u>curl_multi_init() : Retourne un nouveau cURL multiple</u>	442
8.16.19	<u>curl_multi_remove_handle() : Retire un cURL multiple d'un jeu de cURL</u>	442
8.16.20	<u>curl_multi_select() : Retourne les sockets associées à cURL, pour utilisation</u>	443
8.16.21	<u>curl_setopt() : Modifie une option de transfert CURL</u>	443
8.16.22	<u>curl_version() : Retourne la version courante de CURL</u>	451
8.17	<u>Paiement Cybercash</u>	452
8.17.1	<u>Installation</u>	452
8.17.3	<u>cybercash_base64_encode() : Encodage en base64 pour Cybercash</u>	452
8.17.4	<u>cybercash_decr() : Décryptage Cybercash</u>	452
8.17.5	<u>cybercash_encr() : Encryptage Cybercash</u>	452
8.18	<u>Fonctions Crédit Mutuel CyberMUT</u>	453
8.18.1	<u>Introduction</u>	453
8.18.2	<u>Pré-requis</u>	453
8.18.4	<u>cybermut_creeerreponsecm() : Génère le remerciement lors de la confirmation du paiement</u>	453
8.18.5	<u>cybermut_testmac() : Assurez-vous qu'il n'y avait aucune falsification de données dans le message de confirmation reçu</u>	453
8.19	<u>Cyrus IMAP</u>	455
8.19.1	<u>Introduction</u>	455
8.19.2	<u>Installation</u>	455
8.19.3	<u>Constantes pré-définies</u>	455
8.19.5	<u>cyrus_bind() : Ajoute une fonction de callback sur une connexion Cyrus IMAP</u>	455
8.19.6	<u>cyrus_close() : Ferme la connexion à un serveur Cyrus IMAP</u>	456
8.19.7	<u>cyrus_connect() : Se connecte à un serveur Cyrus IMAP</u>	456
8.19.8	<u>cyrus_query() : Envoie une requête à un serveur Cyrus IMAP</u>	456
8.19.9	<u>cyrus_unbind() : Supprime ...</u>	456
8.20	<u>Dates et heures</u>	457
8.20.1	<u>Introduction</u>	457
8.20.2	<u>Pré-requis</u>	457
8.20.3	<u>Installation</u>	457

Sommaire

8 Référence des fonctions

<u>8.20.4 Configuration à l'exécution</u>	457
<u>8.20.5 Configuration à l'exécution</u>	457
<u>8.20.6 Types de ressources</u>	458
<u>8.20.7 Constantes pré-définies</u>	458
<u>8.20.9 date_default_timezone_get() : Récupère le décalage horaire par défaut utilisé par toutes les fonctions date/heure dans un script</u>	459
<u>8.20.10 date_default_timezone_set() : Définit le décalage horaire par défaut utilisé par toutes les fonctions date/heure dans un script</u>	459
<u>8.20.11 date_sunrise() : Retourne l'heure de levé du soleil pour un jour et un endroit donnés</u>	459
<u>8.20.12 date_sunset() : Retourne l'heure de coucher du soleil pour un jour et un endroit donnés</u>	460
<u>8.20.13 date() : Formate une date/heure locale</u>	461
<u>8.20.14 getdate() : Retourne la date/heure</u>	464
<u>8.20.15 gettimeofday() : Retourne l'heure actuelle</u>	465
<u>8.20.16 gmmdate() : Formate une date/heure GMT/CUT</u>	466
<u>8.20.17 gmmktime() : Retourne le timestamp UNIX d'une date GMT</u>	466
<u>8.20.18 gmstrftime() : Formate une date/heure GMT/CUT en fonction des paramétrages locaux</u>	467
<u>8.20.19 idate() : Formate une date/heure locale en tant qu'entier</u>	467
<u>8.20.20 localtime() : Lit l'heure locale</u>	468
<u>8.20.21 microtime() : Retourne le timestamp UNIX actuel avec microsecondes</u>	469
<u>8.20.22 mktime() : Retourne le timestamp UNIX d'une date</u>	470
<u>8.20.23 strftime() : Formate une date/heure locale avec les options locales</u>	470
<u>8.20.24 strtotime() : Analyse une date générée par strftime</u>	473
<u>8.20.25 strtotime() : Transforme un texte anglais en timestamp</u>	473
<u>8.20.26 time() : Retourne le timestamp UNIX actuel</u>	473
<u>8.21 DBA</u>	475
<u>8.21.1 Introduction</u>	475
<u>8.21.2 Pré-requis</u>	475
<u>8.21.3 Installation</u>	476
<u>8.21.4 Configuration à l'exécution</u>	477
<u>8.21.5 Types de ressources</u>	477
<u>8.21.6 Constantes pré-définies</u>	477
<u>8.21.7 Exemples</u>	478
<u>8.21.9 dba_delete() : Efface une ligne dans une base DBA</u>	479
<u>8.21.10 dba_exists() : Vérifie qu'une clé DBA existe</u>	479
<u>8.21.11 dba_fetch() : Lit les données liées à une clé DBA</u>	479
<u>8.21.12 dba_firstkey() : Lit la première clé DBA</u>	479
<u>8.21.13 dba_handlers() : Liste les gestionnaires DBA disponibles</u>	479
<u>8.21.14 dba_insert() : Insère une entrée DBA</u>	480
<u>8.21.15 dba_key_split() : Transforme une représentation de clé DBA par chaîne en une représentation par tableau</u>	480
<u>8.21.16 dba_list() : Liste tous les fichiers de bases de données DBA ouverts</u>	480
<u>8.21.17 dba_nextkey() : Lit la clé DBA suivante</u>	480
<u>8.21.18 dba_open() : Ouvre une base de données DBA</u>	480
<u>8.21.19 dba_optimize() : Optimise une base DBA</u>	480
<u>8.21.20 dba_popen() : Ouvre une connexion persistante à une base de données DBA</u>	480
<u>8.21.21 dba_replace() : Remplace ou insère une ligne DBA</u>	481
<u>8.21.22 dba_sync() : Synchronise une base de données DBA</u>	481

Sommaire

8 Référence des fonctions

<u>8.22 dBase</u>	482
<u>8.22.1 Introduction</u>	482
<u>8.22.2 Installation</u>	482
<u>8.22.3 Configuration à l'exécution</u>	482
<u>8.22.4 Types de ressources</u>	483
<u>8.22.5 Exemples</u>	483
<u>8.22.6 Constantes pré-définies</u>	483
<u>8.22.8 dbase_close() : Ferme une base dBase</u>	483
<u>8.22.9 dbase_create() : Crée une base de données dBase</u>	483
<u>8.22.10 dbase_delete_record() : Efface un enregistrement dans une base dBase</u>	484
<u>8.22.11 dbase_get_header_info() : Récupère des informations d'en-tête d'une base de données dBase</u>	484
<u>8.22.12 dbase_get_record_with_names() : Lit un enregistrement dans une base dBase, sous la forme d'un tableau associatif</u>	484
<u>8.22.13 dbase_get_record() : Lit un enregistrement dans une base dBase</u>	484
<u>8.22.14 dbase_numfields() : Compte le nombre de champs d'une base dBase</u>	484
<u>8.22.15 dbase_numrecords() : Compte le nombre d'enregistrements dans une base dBase</u>	485
<u>8.22.16 dbase_open() : Ouvre une base dBase</u>	485
<u>8.22.17 dbase_pack() : Compacte une base dBase</u>	485
<u>8.22.18 dbase_replace_record() : Remplace un enregistrement dans une base dBase</u>	485
<u>8.23 DBM (Obsolète)</u>	486
<u>8.23.1 Introduction</u>	486
<u>8.23.2 Pré-requis</u>	486
<u>8.23.3 Installation</u>	486
<u>8.23.4 Configuration à l'exécution</u>	486
<u>8.23.5 Types de ressources</u>	486
<u>8.23.6 Constantes pré-définies</u>	486
<u>8.23.7 Exemples</u>	486
<u>8.23.9 dbmclose() : Ferme une base de données dbm</u>	487
<u>8.23.10 dbmdelete() : Efface une valeur</u>	487
<u>8.23.11 dbmexists() : Indique si une valeur existe</u>	487
<u>8.23.12 dbmfetch() : Lit une valeur</u>	487
<u>8.23.13 dbmfirstkey() : Lit la première clé</u>	488
<u>8.23.14 dbminsert() : Insère une valeur</u>	488
<u>8.23.15 dbmnextkey() : Lit la clé suivante</u>	488
<u>8.23.16 dbmopen() : Ouvre une base de données dbm</u>	488
<u>8.23.17 dbmreplace() : Remplace une valeur</u>	489
<u>8.24 Fonctions DB++</u>	490
<u>8.24.1 Introduction</u>	490
<u>8.24.2 Pré-requis</u>	490
<u>8.24.3 Installation</u>	490
<u>8.24.4 Configuration à l'exécution</u>	490
<u>8.24.5 Types de ressources</u>	490
<u>8.24.6 Constantes pré-définies</u>	491
<u>8.24.8 dbplus_aql() : Exécute une requête AQL</u>	493
<u>8.24.9 dbplus_chdir() : Lit/modifie le chemin courant de base</u>	493
<u>8.24.10 dbplus_close() : Ferme une relation</u>	494
<u>8.24.11 dbplus_curr() : Lit la ligne courante dans une relation</u>	494
<u>8.24.12 dbplus_errcode() : Obtenir un message d'erreur</u>	494

Sommaire

8 Référence des fonctions

<u>8.24.13 dbplus_erro()</u> : Lit le code d'erreur généré par la dernière opération.....	494
<u>8.24.14 dbplus_find()</u> : Ajoute une contrainte sur une relation.....	495
<u>8.24.15 dbplus_first()</u> : Lit la première ligne d'une relation.....	495
<u>8.24.16 dbplus_flush()</u> : Écrit toutes les modifications apportées à une relation.....	495
<u>8.24.17 dbplus_freealllocks()</u> : Libère tous les verrous posés par le client.....	496
<u>8.24.18 dbplus_freelock()</u> : Lève un verrou en écriture sur une ligne.....	496
<u>8.24.19 dbplus_freerlocks()</u> : Lève tous les verrous sur les lignes d'une relation.....	496
<u>8.24.20 dbplus_getlock()</u> : Pose un verrou sur une ligne.....	497
<u>8.24.21 dbplus_getunique()</u> : Retourne un identifiant unique pour une relation.....	497
<u>8.24.22 dbplus_info()</u> : Récupère les informations sur une relation.....	497
<u>8.24.23 dbplus_last()</u> : Lit la dernière ligne d'une relation.....	497
<u>8.24.24 dbplus_lockrel()</u> : Pose un verrou en écriture sur une relation.....	498
<u>8.24.25 dbplus_next()</u> : Lit la ligne suivante dans la relation.....	498
<u>8.24.26 dbplus_open()</u> : Ouvre un fichier de relation.....	498
<u>8.24.27 dbplus_prev()</u> : Lit la ligne précédente dans la relation.....	499
<u>8.24.28 dbplus_rchperm()</u> : Modifie les droits sur la relation.....	499
<u>8.24.29 dbplus_rcreate()</u> : Crée une nouvelle relation DB++.....	499
<u>8.24.30 dbplus_rctract()</u> : Crée une copie exacte mais vide d'une relation.....	500
<u>8.24.31 dbplus_rctrlike()</u> : Crée une copie vide d'une relation avec les index.....	500
<u>8.24.32 dbplus_resolve()</u> : Résoud les informations d'hôte pour une relation.....	500
<u>8.24.33 dbplus_restorepos()</u> : Restaure la position.....	501
<u>8.24.34 dbplus_rkeys()</u> : Spécifie la nouvelle clé primaire d'une relation.....	501
<u>8.24.35 dbplus_ropen()</u> : Ouvre un fichier de relation local.....	501
<u>8.24.36 dbplus_rquery()</u> : Accomplit une requête AQL locale.....	501
<u>8.24.37 dbplus_rrename()</u> : Renomme une relation.....	502
<u>8.24.38 dbplus_rsecindex()</u> : Crée un nouvel index secondaire pour une relation.....	502
<u>8.24.39 dbplus_runlink()</u> : Détruit une relation.....	502
<u>8.24.40 dbplus_rzap()</u> : Supprime toutes les lignes d'une relation.....	502
<u>8.24.41 dbplus_savepos()</u> : Sauvegarde la position.....	503
<u>8.24.42 dbplus_setindex()</u> : Définit l'index.....	503
<u>8.24.43 dbplus_setindexbynumber()</u> : Définit l'index par un nombre.....	503
<u>8.24.44 dbplus_sql()</u> : Exécute une requête SQL.....	503
<u>8.24.45 dbplus_tcl()</u> : Exécute du code TCL côté serveur.....	504
<u>8.24.46 dbplus_tremove()</u> : Supprime la ligne courante et retourne la nouvelle ligne courante.....	504
<u>8.24.47 dbplus_undo()</u> : Annule.....	504
<u>8.24.48 dbplus_undoprepere()</u> : Prépare l'annulation.....	504
<u>8.24.49 dbplus_unlockrel()</u> : Libère un verrou en écriture posé sur une relation.....	505
<u>8.24.50 dbplus_unselect()</u> : Retire une contrainte d'une relation.....	505
<u>8.24.51 dbplus_update()</u> : Modifie une ligne dans une relation.....	505
<u>8.24.52 dbplus_xlockrel()</u> : Pose un verrou exclusif sur une relation.....	505
<u>8.24.53 dbplus_xunlockrel()</u> : Lève un verrou exclusif posé sur une relation.....	506
<u>8.25 Fonctions dbx</u>	507
<u>8.25.1 Introduction</u>	507
<u>8.25.2 Pré-requis</u>	507
<u>8.25.3 Installation</u>	507
<u>8.25.4 Configuration à l'exécution</u>	507
<u>8.25.5 Types de ressources</u>	508
<u>8.25.6 Constantes pré-définies</u>	508
<u>8.25.8 dbx_compare()</u> : Compare deux lignes DBX afin de les trier.....	509
<u>8.25.9 dbx_connect()</u> : Ouvre une connexion à une base de données.....	509

Sommaire

8 Référence des fonctions

<u>8.25.10 dbx_error()</u> : Rapporte le message d'erreur du dernier appel de fonction DBX.....	510
<u>8.25.11 dbx_escape_string()</u> : Protège une chaîne de caractères pour l'utiliser dans une requête.....	511
<u>8.25.12 dbx_fetch_row()</u> : Lit une ligne dans un résultat DBX ayant l'option DBX_RESULT_UNBUFFERED activée.....	511
<u>8.25.13 dbx_query()</u> : Envoie une requête et lit tous les résultats DBX.....	512
<u>8.25.14 dbx_sort()</u> : Trie un résultat avec une fonction utilisateur.....	515
<u>8.26 Direct IO</u>	516
<u>8.26.1 Introduction</u>	516
<u>8.26.2 Pré-requis</u>	516
<u>8.26.3 Installation</u>	516
<u>8.26.4 Constantes pré-définies</u>	516
<u>8.26.5 Configuration à l'exécution</u>	517
<u>8.26.6 Types de ressources</u>	517
<u>8.26.8 dio_fcntl()</u> : Exécute une fonction fichier de langage C.....	517
<u>8.26.9 dio_open()</u> : Ouvre un nouveau fichier avec les permissions spécifiés.....	518
<u>8.26.10 dio_read()</u> : Lit les octets dans un fichier.....	518
<u>8.26.11 dio_seek()</u> : Déplace le pointeur interne de fichier.....	518
<u>8.26.12 dio_stat()</u> : Lit des informations sur le fichier.....	518
<u>8.26.13 dio_tcsetattr()</u> : Modifie les attributs du terminal et le taux de baud du port série.....	518
<u>8.26.14 dio_truncate()</u> : Tronque un fichier.....	518
<u>8.26.15 dio_write()</u> : Écrit des données dans le fichier avec la possibilité de tronquer sa longueur.....	519
<u>8.27 Accès aux dossiers</u>	520
<u>8.27.1 Introduction</u>	520
<u>8.27.2 Pré-requis</u>	520
<u>8.27.3 Installation</u>	520
<u>8.27.4 Configuration à l'exécution</u>	520
<u>8.27.5 Types de ressources</u>	520
<u>8.27.6 Constantes pré-définies</u>	520
<u>8.27.7 Voir aussi</u>	520
<u>8.27.9 chroot()</u> : Change le dossier racine.....	521
<u>8.27.10 dir()</u> : Classe de dossier DIR.....	521
<u>8.27.11 closedir()</u> : Ferme le pointeur sur le dossier.....	521
<u>8.27.12 getcwd()</u> : Retourne le dossier de travail courant.....	521
<u>8.27.13 opendir()</u> : Ouvre un dossier, et récupère un pointeur dessus.....	521
<u>8.27.14 readdir()</u> : Lit une entrée du dossier.....	522
<u>8.27.15 rewinddir()</u> : Retourne à la première entrée du dossier.....	522
<u>8.27.16 scandir()</u> : Liste les fichiers et dossiers dans un dossier.....	522
<u>8.28 Fonctions DOM</u>	523
<u>8.28.1 Introduction</u>	523
<u>8.28.2 Installation</u>	523
<u>8.28.3 Classes pré-définies</u>	523
<u>8.28.4 Exemples</u>	532
<u>8.28.5 Constantes pré-définies</u>	532
<u>8.28.7 DOMAttr->isId()</u> : Vérifie si l'attribut est un identifiant défini.....	536
<u>8.28.8 DOMCharacterData->appendData()</u> : Ajoute la chaîne à la fin des données dans le noeud.....	536
<u>8.28.9 DOMCharacterData->deleteData()</u> : Supprime une sous-chaîne dans le	

Sommaire

8 Référence des fonctions

<u>noeud</u>	536
<u>8.28.10 DOMCharacterData->insertData()</u> : Insère une chaîne à la position spécifiée d'unité 16-bit.....	536
<u>8.28.11 DOMCharacterData->replaceData()</u> : Remplace une sous-chaîne dans le noeud DOMCharacterData node.....	536
<u>8.28.12 DOMCharacterData->substringData()</u> : Extrait un morceaux de données dans le noeud.....	537
<u>8.28.13 DOMComment->construct()</u> : Crée un nouvel objet DOMComment.....	537
<u>8.28.14 DOMDocument->construct()</u> : Crée un nouvel objet DOMDocument.....	537
<u>8.28.15 DOMDocument->createAttribute()</u> : Crée un nouvel attribut.....	537
<u>8.28.16 DOMDocument->createAttributeNS()</u> : Crée un nouvel attribut avec un espace de noms associé.....	537
<u>8.28.17 DOMDocument->createCDATASection()</u> : Crée un nouveau noeud cdata.....	537
<u>8.28.18 DOMDocument->createComment()</u> : Crée un nouveau noeud de commentaire.....	538
<u>8.28.19 DOMDocument->createDocumentFragment()</u> : Crée un nouveau fragment de document.....	538
<u>8.28.20 DOMDocument->createElement()</u> : Crée un nouveau noeud.....	538
<u>8.28.21 DOMDocument->createElementNS()</u> : Crée un nouveau noeud avec un espace de noms associé.....	538
<u>8.28.22 DOMDocument->createEntityReference()</u> : Crée un nouveau noeud de référence d'entité.....	538
<u>8.28.23 DOMDocument->createProcessingInstruction()</u> : Crée un nouveau noeud PI.....	538
<u>8.28.24 DOMDocument->createTextNode()</u> : Crée un nouveau noeud de texte.....	539
<u>8.28.25 DOMDocument->getElementById()</u> : Cherche un élément avec un certain identifiant.....	539
<u>8.28.26 DOMDocument->getElementsByTagName()</u> : Cherche tous les éléments quit ont le nom de balise donné.....	539
<u>8.28.27 DOMDocument->getElementsByTagNameNS()</u> : Recherche tous les éléments avec un nom de balise donné dans un espace de noms spécifié.....	539
<u>8.28.28 DOMDocument->importNode()</u> : Importe un noeud dans le document courant.....	539
<u>8.28.29 DOMDocument->load()</u> : Charge du XML depuis un fichier.....	540
<u>8.28.30 DOMDocument->loadHTML()</u> : Charge du HTML à partir d'une chaîne de caractères.....	540
<u>8.28.31 DOMDocument->loadHTMLFile()</u> : Charge du HTML à partir d'un fichier.....	540
<u>8.28.32 DOMDocument->loadXML()</u> : Charge du XML depuis une chaîne de caractères.....	540
<u>8.28.33 DOMDocument->normalize()</u> : Normalise le document.....	541
<u>8.28.34 DOMDocument->relaxNGValidate()</u> : Effectue une validation relaxNG sur le document.....	541
<u>8.28.35 DOMDocument->relaxNGValidateSource()</u> : Effectue une validation relaxNG sur le document.....	541
<u>8.28.36 DOMDocument->save()</u> : Sauvegarde l'arbre interne XML dans un fichier.....	541
<u>8.28.37 DOMDocument->saveHTML()</u> : Sauvegarde le document interne dans une chaîne en utilisant un formatage HTML.....	541
<u>8.28.38 DOMDocument->saveHTMLFile()</u> : Sauvegarde un document interne dans un fichier en utilisant un formatage HTML.....	541
<u>8.28.39 DOMDocument->saveXML()</u> : Sauvegarde l'arbre interne XML dans une	

Sommaire

8 Référence des fonctions

<u>chaîne de caractères</u>	542
<u>8.28.40 DOMDocument->schemaValidate()() : Valide un document selon un schéma</u>	542
<u>8.28.41 DOMDocument->schemaValidateSource()() : Valide un document selon un schéma</u>	542
<u>8.28.42 DOMDocument->validate()() : Valide un document en se basant sur sa DTD</u>	542
<u>8.28.43 DOMDocument->xinclude()() : Remplace les XIncludes dans un objet DOMDocument</u>	542
<u>8.28.44 DOMELEMENT->construct()() : Crée un nouvel objet DOMELEMENT</u>	542
<u>8.28.45 DOMELEMENT->getAttribute()() : Retourne la valeur d'un attribut</u>	543
<u>8.28.46 DOMELEMENT->getAttributeNode()() : Retourne le noeud d'attribut</u>	543
<u>8.28.47 DOMELEMENT->getAttributeNodeNS()() : Retourne le noeud d'attribut</u>	543
<u>8.28.48 DOMELEMENT->getAttributeNS()() : Retourne la valeur de l'attribut</u>	543
<u>8.28.49 DOMELEMENT->getElementsByTagName()() : Retourne les éléments par leur nom de balise</u>	543
<u>8.28.50 DOMELEMENT->getElementsByTagNameNS()() : Récupère les éléments par leur espace de noms et leur localName</u>	543
<u>8.28.51 DOMELEMENT->hasAttribute()() : Vérifie si un attribut existe</u>	544
<u>8.28.52 DOMELEMENT->hasAttributeNS()() : Vérifie si un attribut existe</u>	544
<u>8.28.53 DOMELEMENT->removeAttribute()() : Efface un attribut</u>	544
<u>8.28.54 DOMELEMENT->removeAttributeNode()() : Efface un attribut</u>	544
<u>8.28.55 DOMELEMENT->removeAttributeNS()() : Efface un attribut</u>	544
<u>8.28.56 DOMELEMENT->setAttribute()() : Ajoute un nouvel attribut</u>	544
<u>8.28.57 DOMELEMENT->setAttributeNode()() : Ajoute un nouvel attribut à l'élément</u>	544
<u>8.28.58 DOMELEMENT->setAttributeNodeNS()() : Ajoute un nouvel attribut à l'élément</u>	545
<u>8.28.59 DOMELEMENT->setAttributeNS()() : Ajoute un nouvel attribut</u>	545
<u>8.28.60 DOMEntityReference->construct()() : Crée un nouvel objet DOMEntityReference</u>	545
<u>8.28.61 DOMImplementation->construct()() : Crée un nouvel objet DOMImplementation</u>	545
<u>8.28.62 DOMImplementation->createDocument()() : Crée un objet DOM Document du type spécifié avec ses éléments</u>	545
<u>8.28.63 DOMImplementation->createDocumentType()() : Crée un objet DOMDocumentType vide</u>	545
<u>8.28.64 DOMImplementation->hasFeature()() : Test si l'implémentation DOM implémente une fonctionnalité spécifique</u>	546
<u>8.28.65 DOMNamedNodeMap->getNamedItem()() : Retourne un noeud spécifié par nom</u>	546
<u>8.28.66 DOMNamedNodeMap->getNamedItemNS()() : Retourne un noeud spécifié par son nom local et son espace de nom</u>	546
<u>8.28.67 DOMNamedNodeMap->item()() : Retourne un noeud selon son index</u>	546
<u>8.28.68 DOMNode->appendChild()() : Ajoute un nouveau fils à la fin des fils</u>	546
<u>8.28.69 DOMNode->cloneNode()() : Clone un noeud</u>	547
<u>8.28.70 DOMNode->hasAttributes()() : Vérifie si le noeud possède un attribut</u>	547
<u>8.28.71 DOMNode->hasChildNodes()() : Vérifie si le noeud possède des enfants</u>	547
<u>8.28.72 DOMNode->insertBefore()() : Ajoute un nouveau fils à la fin des enfants</u>	547
<u>8.28.73 DOMNode->isSameNode()() : Indique si deux noeuds sont identiques</u>	547
<u>8.28.74 DOMNode->isSupported()() : Vérifie si la fonctionnalité est disponible pour la version spécifiée</u>	547

Sommaire

8 Référence des fonctions

<u>8.28.75 DOMNode->lookupNamespaceURI()</u> : Retourne l'URI de l'espace de noms selon le préfixe.....	548
<u>8.28.76 DOMNode->lookupPrefix()</u> : Retourne le préfixe de l'espace de noms selon l'URI de l'espace de noms.....	548
<u>8.28.77 DOMNode->normalize()</u> : Normalise le noeud.....	548
<u>8.28.78 DOMNode->removeChild()</u> : Supprime un fils de la liste des enfants.....	548
<u>8.28.79 DOMNode->replaceChild()</u> : Remplace un fils.....	548
<u>8.28.80 DOMNodeList->item()</u> : Retourne un noeud spécifié par son index.....	548
<u>8.28.81 DOMProcessingInstruction->construct()</u> : Crée un nouvel objet DOMProcessingInstruction.....	549
<u>8.28.82 DOMText->construct()</u> : Crée un nouvel objet DOMText.....	549
<u>8.28.83 DOMText->isWhitespaceInElementContent()</u> : Indique si ce noeud de texte contient des espaces blancs.....	549
<u>8.28.84 DOMText->splitText()</u> : Coupe le noeud en deux noeuds à l'endroit spécifié.....	549
<u>8.28.85 DOMXPath->construct()</u> : Crée un nouvel objet DOMXPath.....	549
<u>8.28.86 DOMXPath->evaluate()</u> : Evalue une expression XPath donnée et retourne un résultat écrit si possible.....	549
<u>8.28.87 DOMXPath->query()</u> : Evalue l'expression XPath donnée.....	550
<u>8.28.88 DOMXPath->registerNamespace()</u> : Enregistre l'espace de noms avec l'objet DOMXPath.....	550
<u>8.28.89 dom_import_simplexml()</u> : Transforme un objet DOMELEMENT en un objet SimpleXMLElement.....	550
<u>8.29 Fonctions DOM XML</u>	551
<u>8.29.1 Introduction</u>	551
<u>8.29.2 Pré-requis</u>	551
<u>8.29.3 Installation</u>	551
<u>8.29.4 Fonctions dépréciées</u>	552
<u>8.29.5 Constantes pré-définies</u>	552
<u>8.29.6 Classes</u>	553
<u>8.29.7 Exemples</u>	556
<u>8.29.9 DomAttribute->set_value()</u> : Spécifie une valeur à un attribut.....	558
<u>8.29.10 DomAttribute->specified()</u> : Vérifie si un attribut est spécifié.....	558
<u>8.29.11 DomAttribute->value()</u> : Retourne la valeur d'un attribut.....	558
<u>8.29.12 DomDocument->add_root()</u> : Ajoute un noeud.....	559
<u>8.29.13 DomDocument->create_attribute()</u> : Crée un nouvel attribut.....	559
<u>8.29.14 DomDocument->create_cdata_section()</u> : Crée un nouveau noeud CDATA.....	559
<u>8.29.15 DomDocument->create_comment()</u> : Crée un nouveau noeud de commentaire.....	559
<u>8.29.16 DomDocument->create_element_ns()</u> : Crée un nouvel élément avec un espace de noms associé.....	560
<u>8.29.17 DomDocument->create_element()</u> : Crée un nouveau noeud élément.....	560
<u>8.29.18 DomDocument->create_entity_reference()</u> : Crée une entité référence.....	560
<u>8.29.19 DomDocument->create_processing_instruction()</u> : Crée un nouveau noeud de traitement d'instructions (PI).....	561
<u>8.29.20 DomDocument->create_text_node()</u> : Crée un nouveau noeud de texte.....	561
<u>8.29.21 DomDocument->doctype()</u> : Retourne le type de document.....	561
<u>8.29.22 DomDocument->document_element()</u> : Retourne le noeud racine d'un document DOM XML.....	562
<u>8.29.23 DomDocument->dump_file()</u> : Convertit un document DOM XML en	

Sommaire

8 Référence des fonctions

<u>fichier</u>	562
<u>8.29.24 DomDocument->dump_mem()</u> : Convertit l'arbre XML interne en une chaîne de caractères.....	563
<u>8.29.25 DomDocument->get_element_by_id()</u> : Recherche un élément avec son identifiant.....	563
<u>8.29.26 DomDocument->get_elements_by_tagname()</u> : Retourne un tableau avec noeuds pour le nom de tag donné dans le document ou un tableau vide si non trouvé.....	564
<u>8.29.27 DomDocument->html_dump_mem()</u> : Convertit l'arbre XML interne en une chaîne de caractères au format HTML.....	564
<u>8.29.28 DomDocument->xinclude()</u> : Remplace XIncludes dans un objet DomDocument.....	564
<u>8.29.29 DomDocumentType->entities()</u> : Retourne la liste des entités.....	565
<u>8.29.30 DomDocumentType->internal_subset()</u> : Retourne le sous ensemble interne (internal subset).....	565
<u>8.29.31 DomDocumentType->name()</u> : Retourne le nom du type de document.....	565
<u>8.29.32 DomDocumentType->notations()</u> : Retourne la liste des notations.....	566
<u>8.29.33 DomDocumentType->public_id()</u> : Retourne l'identifiant public du type de document.....	566
<u>8.29.34 DomDocumentType->system_id()</u> : Retourne l'identifiant système du type de document.....	566
<u>8.29.35 DomElement->get_attribute_node()</u> : Retourne la valeur d'un attribut.....	566
<u>8.29.36 DomElement->get_attribute()</u> : Retourne la valeur d'un attribut.....	566
<u>8.29.37 DomElement->get_elements_by_tagname()</u> : Lit les éléments par nom de balise.....	566
<u>8.29.38 DomElement->has_attribute()</u> : Vérifie si un attribut existe.....	567
<u>8.29.39 DomElement->remove_attribute()</u> : Supprime un attribut.....	567
<u>8.29.40 DomElement->set_attribute()</u> : Définit la valeur d'un attribut.....	567
<u>8.29.41 DomElement->tagname()</u> : Retourne le nom de l'élément courant.....	567
<u>8.29.42 DomNode->add_namespace()</u> : Ajoute une déclaration d'espace de nom à un noeud.....	567
<u>8.29.43 DomNode->append_child()</u> : Ajoute un nouveau fils à la fin des enfants.....	567
<u>8.29.44 DomNode->append_sibling()</u> : Ajoute un frère à un noeud.....	567
<u>8.29.45 DomNode->attributes()</u> : Retourne la liste des attributs.....	568
<u>8.29.46 DomNode->child_nodes()</u> : Retourne les fils d'un noeud.....	568
<u>8.29.47 DomNode->clone_node()</u> : Clone un noeud.....	568
<u>8.29.48 DomNode->dump_node()</u> : Convertit un noeud en chaîne.....	568
<u>8.29.49 DomNode->first_child()</u> : Retourne le premier fils du noeud.....	569
<u>8.29.50 DomNode->get_content()</u> : Retourne le contenu du noeud.....	569
<u>8.29.51 DomNode->has_attributes()</u> : Vérifie si un noeud a des attributs.....	569
<u>8.29.52 DomNode->has_child_nodes()</u> : Vérifie si un noeud a des fils.....	569
<u>8.29.53 DomNode->insert_before()</u> : Insère un nouveau noeud fils.....	570
<u>8.29.54 DomNode->is_blank_node()</u> : Vérifie si un noeud est blanc.....	570
<u>8.29.55 DomNode->last_child()</u> : Retourne le dernier fils du noeud.....	570
<u>8.29.56 DomNode->next_sibling()</u> : Retourne le noeud frère suivant.....	571
<u>8.29.57 DomNode->node_name()</u> : Retourne le nom du noeud.....	571
<u>8.29.58 DomNode->node_type()</u> : Retourne le type de noeud.....	572
<u>8.29.59 DomNode->node_value()</u> : Retourne la valeur d'un noeud.....	572
<u>8.29.60 DomNode->owner_document()</u> : Retourne le document auquel appartient ce noeud DOM XML.....	573
<u>8.29.61 DomNode->parent_node()</u> : Retourne le père d'un noeud DOM XML.....	573

Sommaire

8 Référence des fonctions

<u>8.29.62 DomNode->prefix()</u> : Retourne le préfixe d'espace de nom d'un noeud.....	573
<u>8.29.63 DomNode->previous_sibling()</u> : Retourne le frère précédent d'un noeud.....	574
<u>8.29.64 DomNode->remove_child()</u> : Supprime un fils de la liste des noeuds fils.....	574
<u>8.29.65 DomNode->replace_child()</u> : Remplace un noeud fils.....	574
<u>8.29.66 DomNode->replace_node()</u> : Remplace un noeud DomXML.....	575
<u>8.29.67 DomNode->set_content()</u> : Modifie le contenu d'un noeud.....	575
<u>8.29.68 DomNode->set_name()</u> : Modifie le nom d'un noeud.....	575
<u>8.29.69 DomNode->set_namespace()</u> : Modifie l'espace de noms d'un noeud.....	575
<u>8.29.70 DomNode->unlink_node()</u> : Efface un noeud.....	576
<u>8.29.71 DomProcessingInstruction->data()</u> : Retourne les données du noeud l'instruction en cours.....	576
<u>8.29.72 DomProcessingInstruction->target()</u> : Retourne la cible du noeud de l'instruction en cours.....	576
<u>8.29.73 domxml_new_doc()</u> : Crée un document XML vide.....	576
<u>8.29.74 domxml_open_file()</u> : Crée un objet DOM à partir d'un fichier XML.....	576
<u>8.29.75 domxml_open_mem()</u> : Crée un objet DOM pour un document XML.....	576
<u>8.29.76 domxml_version()</u> : Lit le numéro de version de la bibliothèque XML.....	576
<u>8.29.77 domxml_xmltree()</u> : Crée un arbre d'objets PHP à partir d'un document XML.....	577
<u>8.29.78 domxml_xslt_stylesheet_doc()</u> : Crée un objet DomXsltStylesheet à partir d'un objet DomDocument.....	577
<u>8.29.79 domxml_xslt_stylesheet_file()</u> : Crée un objet DomXsltStylesheet à partir d'un document XSL dans un fichier.....	577
<u>8.29.80 domxml_xslt_stylesheet()</u> : Crée un objet DomXsltStylesheet à partir d'un document XSL dans une chaîne.....	577
<u>8.29.81 domxml_xslt_version()</u> : Lit le numéro de version de la bibliothèque XSLT.....	577
<u>8.29.82 DomXsltStylesheet->process()</u> : Applique une transformation XSLT à un objet domdocument.....	578
<u>8.29.83 DomXsltStylesheet->result_dump_file()</u> : Écrit le résultat d'un transformation XSLT dans un fichier.....	578
<u>8.29.84 DomXsltStylesheet->result_dump_mem()</u> : Écrit le résultat d'un transformation XSLT dans une chaîne.....	578
<u>8.29.85 xpath_eval_expression()</u> : Calcule un chemin XPath à partir d'une chaîne.....	578
<u>8.29.86 xpath_eval()</u> : Calcule un chemin XPath à partir d'une chaîne.....	579
<u>8.29.87 xpath_new_context()</u> : Crée un nouveau contexte xpath.....	579
<u>8.29.88 xpath_register_ns_auto()</u> : Sauvegarde l'espace de nom donné dans le contexte XPath passé.....	579
<u>8.29.89 xpath_register_ns()</u> : Sauvegarde l'espace de nom donné dans le contexte XPath passé.....	580
<u>8.29.90 xptr_eval()</u> : Calcul un chemin XPtr à partir d'une chaîne.....	580
<u>8.29.91 xptr_new_context()</u> : Crée un nouveau contexte XPath.....	580
<u>8.30 Fonctions .NET</u>	581
<u>8.30.1 Introduction</u>	581
<u>8.31 Gestion des erreurs</u>	582
<u>8.31.1 Introduction</u>	582
<u>8.31.2 Pré-requis</u>	582
<u>8.31.3 Installation</u>	582
<u>8.31.4 Configuration à l'exécution</u>	582
<u>8.31.5 Constantes pré-définies</u>	585
<u>8.31.6 Exemples</u>	586
<u>8.31.7 Voir aussi</u>	588

Sommaire

8 Référence des fonctions

<u>8.31.9 debug_print_backtrace()</u> : Affiche une backtrace.....	588
<u>8.31.10 error_log()</u> : Stocke un message d'erreur.....	588
<u>8.31.11 error_reporting()</u> : Fixe le niveau de rapport d'erreurs PHP.....	589
<u>8.31.12 restore_error_handler()</u> : Réactive l'ancienne fonction de gestion des erreurs.....	589
<u>8.31.13 restore_exception_handler()</u> : Réactive l'ancienne fonction de gestion d'exceptions.....	589
<u>8.31.14 set_error_handler()</u> : Spécifie une fonction utilisateur comme gestionnaire d'erreurs.....	589
<u>8.31.15 set_exception_handler()</u> : Définit une fonction utilisateur de gestion d'exceptions.....	590
<u>8.31.16 trigger_error()</u> : Déclenche une erreur utilisateur.....	590
<u>8.31.17 user_error()</u> : Alias de trigger_error.....	590
<u>8.32 Exécution de programmes externes</u>	591
<u>8.32.1 Introduction</u>	591
<u>8.32.2 Pré-requis</u>	591
<u>8.32.3 Installation</u>	591
<u>8.32.4 Configuration à l'exécution</u>	591
<u>8.32.5 Types de ressources</u>	591
<u>8.32.6 Constantes pré-définies</u>	591
<u>8.32.7 Voir aussi</u>	591
<u>8.32.9 escapeshellcmd()</u> : Protège les caractères spéciaux du Shell.....	592
<u>8.32.10 exec()</u> : Exécute un programme externe.....	592
<u>8.32.11 passthru()</u> : Exécute un programme externe et affiche le résultat brut.....	592
<u>8.32.12 proc_close()</u> : Ferme un processus ouvert par proc_open et retourne le code de sortie.....	592
<u>8.32.13 proc_get_status()</u> : Lit les informations concernant un processus ouvert par proc_open.....	593
<u>8.32.14 proc_nice()</u> : Change la priorité d'exécution du processus courant.....	593
<u>8.32.15 proc_open()</u> : Exécute une commande et ouvre les pointeurs de fichiers pour les entrées / sorties.....	593
<u>8.32.16 proc_terminate()</u> : Termine un processus ouvert par proc_open.....	593
<u>8.32.17 shell_exec()</u> : Exécute une commande via le Shell et retourne le résultat sous forme de chaîne.....	594
<u>8.32.18 system()</u> : Exécute un programme externe et affiche le résultat.....	594
<u>8.33 EXIF</u>	595
<u>8.33.1 Introduction</u>	595
<u>8.33.2 Pré-requis</u>	595
<u>8.33.3 Installation</u>	595
<u>8.33.4 Configuration à l'exécution</u>	595
<u>8.33.5 Types de ressources</u>	596
<u>8.33.6 Constantes pré-définies</u>	596
<u>8.33.8 exif_read_data()</u> : Lit les en-têtes EXIF dans les images JPEG ou TIFF.....	597
<u>8.33.9 exif_tagname()</u> : Lit le nom d'en-tête EXIF d'un index.....	597
<u>8.33.10 exif_thumbnail()</u> : Récupère la miniature d'une image TIFF ou JPEG.....	597
<u>8.33.11 read_exif_data()</u> : Alias de exif_read_data.....	598
<u>8.34 Fonctions de monitoring d'altération de fichiers</u>	599
<u>8.34.1 Introduction</u>	599
<u>8.34.2 Pré-requis</u>	599
<u>8.34.3 Installation</u>	599
<u>8.34.4 Configuration à l'exécution</u>	599

Sommaire

8 Référence des fonctions

<u>8.34.5 Types de ressources</u>	599
<u>8.34.6 Constantes pré-définies</u>	599
<u>8.34.8 fam_close() : Ferme la connexion FAM</u>	600
<u>8.34.9 fam_monitor_collection() : Monitore les changements d'une collection de fichiers dans un dossier</u>	600
<u>8.34.10 fam_monitor_directory() : Monitore un dossier pour les changements</u>	601
<u>8.34.11 fam_monitor_file() : Monitore un fichier régulier pour les changements</u>	601
<u>8.34.12 fam_next_event() : Retourne le prochain événement FAM en attente</u>	601
<u>8.34.13 fam_open() : Ouvre une connexion vers le démon FAM</u>	601
<u>8.34.14 fam_pending() : Vérifie si des événements FAM sont disponibles</u>	602
<u>8.34.15 fam_resume_monitor() : Reprend un monitoring suspendu</u>	602
<u>8.34.16 fam_suspend_monitor() : Suspend temporairement le monitoring</u>	602
<u>8.35 FrontBase</u>	603
<u>8.35.1 Introduction</u>	603
<u>8.35.2 Pré-requis</u>	603
<u>8.35.3 Installation</u>	603
<u>8.35.4 Configuration à l'exécution</u>	603
<u>8.35.5 Types de ressources</u>	604
<u>8.35.6 Constantes pré-définies</u>	604
<u>8.35.8 fbsql_autocommit() : Active ou désactive la validation automatique</u>	605
<u>8.35.9 fbsql_blob_size() : Retourne la taille d'un BLOB</u>	606
<u>8.35.10 fbsql_change_user() : Change le nom d'utilisateur de la session active</u>	606
<u>8.35.11 fbsql_clob_size() : Retourne la taille d'un CLOB</u>	606
<u>8.35.12 fbsql_close() : Ferme la connexion FrontBase</u>	606
<u>8.35.13 fbsql_commit() : Valide une transaction</u>	607
<u>8.35.14 fbsql_connect() : Ouvre une connexion à un serveur FrontBase</u>	607
<u>8.35.15 fbsql_create_blob() : Crée un BLOB</u>	607
<u>8.35.16 fbsql_create_clob() : Crée un CLOB</u>	608
<u>8.35.17 fbsql_create_db() : Crée une base de données</u>	608
<u>8.35.18 fbsql_data_seek() : Déplace le pointeur interne de résultat FBSQL</u>	609
<u>8.35.19 fbsql_database_password() : Modifie/lit le mot de passe dans une base FBSQL</u>	609
<u>8.35.20 fbsql_database() : Lit ou définit le nom de la base de données utilisée avec une connexion</u>	610
<u>8.35.21 fbsql_db_query() : Envoie une requête à la base FrontBase</u>	610
<u>8.35.22 fbsql_db_status() : Lit le statut courant d'une base de données</u>	610
<u>8.35.23 fbsql_drop_db() : Supprime une base de données FrontBase</u>	611
<u>8.35.24 fbsql_errno() : Retourne le code d'erreur FrontBase</u>	611
<u>8.35.25 fbsql_error() : Retourne le message d'erreur FrontBase</u>	611
<u>8.35.26 fbsql_fetch_array() : Lit toute une ligne de résultat dans un tableau</u>	612
<u>8.35.27 fbsql_fetch_assoc() : Lit toute une ligne de résultat dans un tableau associatif</u>	612
<u>8.35.28 fbsql_fetch_field() : Lit des informations sur une colonne dans un résultat, et retourne un objet</u>	613
<u>8.35.29 fbsql_fetch_lengths() : Lit la taille de chaque colonne d'un résultat</u>	614
<u>8.35.30 fbsql_fetch_object() : Lit une ligne de résultat sous forme d'objet</u>	614
<u>8.35.31 fbsql_fetch_row() : Lit une ligne de résultat sous forme de tableau numérique</u>	615
<u>8.35.32 fbsql_field_flags() : Lit les options associées à une colonne de résultat</u>	615
<u>8.35.33 fbsql_field_len() : Retourne la taille d'un champ</u>	615
<u>8.35.34 fbsql_field_name() : Lit le nom d'un champ</u>	615

Sommaire

8 Référence des fonctions

<u>8.35.35 fbsql_field_seek()</u> : Déplace le pointeur de résultat.....	616
<u>8.35.36 fbsql_field_table()</u> : Lit le nom de la table d'origine d'un champ.....	616
<u>8.35.37 fbsql_field_type()</u> : Lit le type d'une colonne.....	616
<u>8.35.38 fbsql_free_result()</u> : Libère le résultat de la mémoire.....	617
<u>8.35.39 fbsql_get_autostart_info()</u> : Pas de documentation.....	617
<u>8.35.40 fbsql_hostname()</u> : Lit ou définit le nom de l'hôte à utiliser avec une connexion.....	617
<u>8.35.41 fbsql_insert_id()</u> : Lit le dernier identifiant généré par une requête INSERT.....	617
<u>8.35.42 fbsql_list_dbs()</u> : Liste les bases de données FBSQL.....	618
<u>8.35.43 fbsql_list_fields()</u> : Liste les champs d'un résultat FrontBase.....	618
<u>8.35.44 fbsql_list_tables()</u> : Liste les tables dans une base de données FrontBase.....	619
<u>8.35.45 fbsql_next_result()</u> : Déplace le pointeur interne vers le résultat suivant.....	619
<u>8.35.46 fbsql_num_fields()</u> : Lit le nombre de champs dans un résultat.....	619
<u>8.35.47 fbsql_num_rows()</u> : Lit le nombre de lignes dans un résultat.....	620
<u>8.35.48 fbsql_password()</u> : Lit ou définit le mot de passe à utiliser avec une connexion.....	620
<u>8.35.49 fbsql_pconnect()</u> : Ouvre une connexion persistante à un serveur FrontBase.....	620
<u>8.35.50 fbsql_query()</u> : Exécute une requête sur un serveur FrontBase.....	621
<u>8.35.51 fbsql_read_blob()</u> : Lit un BLOB dans une base de données.....	622
<u>8.35.52 fbsql_read_clob()</u> : Lit un CLOB dans une base de données.....	622
<u>8.35.53 fbsql_result()</u> : Lit des données dans un résultat.....	623
<u>8.35.54 fbsql_rollback()</u> : Annule une transaction FBSQL.....	623
<u>8.35.55 fbsql_select_db()</u> : Sélectionne une base de données FrontBase.....	623
<u>8.35.56 fbsql_set_lob_mode()</u> : Modifie le mode de lecture des LOB.....	624
<u>8.35.57 fbsql_set_password()</u> : Change le mot de passe de l'utilisateur spécifié.....	624
<u>8.35.58 fbsql_set_transaction()</u> : Configure le verrouillage et l'isolation de transactions.....	625
<u>8.35.59 fbsql_start_db()</u> : Démarre une base de données FBSQL.....	625
<u>8.35.60 fbsql_stop_db()</u> : Stoppe une base de données FBSQL.....	625
<u>8.35.61 fbsql_tablename()</u> : Lit le nom de la table d'un champ.....	625
<u>8.35.62 fbsql_username()</u> : Lit ou définit le nom de login à utiliser avec une connexion.....	626
<u>8.35.63 fbsql_warnings()</u> : Active ou désactive les alertes FrontBase.....	626
<u>8.36 Forms Data Format</u>	627
<u>8.36.1 Introduction</u>	627
<u>8.36.2 Pré-requis</u>	627
<u>8.36.3 Installation</u>	627
<u>8.36.4 Configuration à l'exécution</u>	628
<u>8.36.5 Types de ressources</u>	628
<u>8.36.6 Constantes pré-définies</u>	628
<u>8.36.7 Exemples</u>	629
<u>8.36.9 fdf_add_template()</u> : Ajoute un template dans le document FDF.....	630
<u>8.36.10 fdf_close()</u> : Ferme un document FDF.....	630
<u>8.36.11 fdf_create()</u> : Crée un nouveau document FDF.....	630
<u>8.36.12 fdf_enum_values()</u> : Appelle une fonction utilisateur à chaque valeur FDF.....	631
<u>8.36.13 fdf_erro()</u> : Retourne le code d'erreur de la dernière opération FDF.....	631
<u>8.36.14 fdf_error()</u> : Retourne le message d'erreur FDF.....	631
<u>8.36.15 fdf_get_ap()</u> : Lit l'apparence d'un champ.....	631
<u>8.36.16 fdf_get_attachment()</u> : Extrait un fichier intégré dans un document FDF.....	631
<u>8.36.17 fdf_get_encoding()</u> : Lit la valeur de la clé /Encoding.....	632

Sommaire

8 Référence des fonctions

<u>8.36.18 fdf_get_file()</u> : Lit la valeur de la clé /F.....	632
<u>8.36.19 fdf_get_flags()</u> : Lit les attributs d'un champ FDF.....	632
<u>8.36.20 fdf_get_opt()</u> : Lit une valeur dans un tableau de valeur d'un champ FDF.....	633
<u>8.36.21 fdf_get_status()</u> : Lit la valeur de la clé /STATUS.....	633
<u>8.36.22 fdf_get_value()</u> : Retourne la valeur d'un champ FDF.....	633
<u>8.36.23 fdf_get_version()</u> : Lit le numéro de version de l'API FDF.....	633
<u>8.36.24 fdf_header()</u> : Emet les en-têtes HTTP spécifiques à FDF.....	633
<u>8.36.25 fdf_next_field_name()</u> : Lit le nom du champ FDF suivant.....	634
<u>8.36.26 fdf_open_string()</u> : Lit un document FDF à partir d'une chaîne de caractères.....	634
<u>8.36.27 fdf_open()</u> : Ouvre un document FDF.....	634
<u>8.36.28 fdf_remove_item()</u> : Configure le cadre FDF de destination pour le formulaire.....	635
<u>8.36.29 fdf_save_string()</u> : Retourne un document FDF sous forme de chaîne.....	635
<u>8.36.30 fdf_save()</u> : Sauvegarde un document FDF.....	636
<u>8.36.31 fdf_set_ap()</u> : Fixe l'apparence d'un champ FDF.....	636
<u>8.36.32 fdf_set_encoding()</u> : Modifie l'encodage des caractères.....	636
<u>8.36.33 fdf_set_file()</u> : Crée un document PDF pour y afficher des données FDF.....	636
<u>8.36.34 fdf_set_flags()</u> : Modifie une option d'un champ.....	637
<u>8.36.35 fdf_set_javascript_action()</u> : Modifie l'action javascript d'un champ.....	637
<u>8.36.36 fdf_set_on_import_javascript()</u> : Ajoute du code Javascript à être exécuté lorsque Acrobat ouvre un FDF.....	637
<u>8.36.37 fdf_set_opt()</u> : Modifie une option d'un champ.....	637
<u>8.36.38 fdf_set_status()</u> : Fixe la valeur de la clé /STATUS.....	638
<u>8.36.39 fdf_set_submit_form_action()</u> : Modifie l'action d'un formulaire.....	638
<u>8.36.40 fdf_set_target_frame()</u> : Configure le cadre de destination pour l'affichage du formulaire.....	638
<u>8.36.41 fdf_set_value()</u> : Modifie la valeur d'un champ FDF.....	638
<u>8.36.42 fdf_set_version()</u> : Modifie le numéro de version du fichier FDF.....	639
<u>8.37 Fonctions filePro</u>	640
<u>8.37.1 Introduction</u>	640
<u>8.37.2 Installation</u>	640
<u>8.37.4 filepro_fieldname()</u> : Retourne le nom d'un champ filePro.....	640
<u>8.37.5 filepro_fieldtype()</u> : Retourne le type d'un champ filePro.....	640
<u>8.37.6 filepro_fieldwidth()</u> : Retourne la taille d'un champ filePro.....	640
<u>8.37.7 filepro_retrieve()</u> : Retourne la valeur d'un champ filePro.....	641
<u>8.37.8 filepro_rowcount()</u> : Retourne le nombre de lignes dans une base filePro.....	641
<u>8.37.9 filepro()</u> : Lit et vérifie un fichier.....	641
<u>8.38 Système de fichiers</u>	642
<u>8.38.1 Introduction</u>	642
<u>8.38.2 Pré-requis</u>	642
<u>8.38.3 Installation</u>	642
<u>8.38.4 Configuration à l'exécution</u>	642
<u>8.38.5 Types de ressources</u>	643
<u>8.38.6 Constantes pré-définies</u>	643
<u>8.38.7 Voir aussi</u>	643
<u>8.38.9 chgrp()</u> : Change le groupe propriétaire d'un fichier.....	645
<u>8.38.10 chmod()</u> : Change le mode du fichier.....	646
<u>8.38.11 chown()</u> : Change le groupe propriétaire du fichier.....	647
<u>8.38.12 clearstatcache()</u> : Efface le cache de stat.....	647
<u>8.38.13 copy()</u> : Copie un fichier.....	647

Sommaire

8 Référence des fonctions

<u>8.38.14 delete()</u> : Effacer.....	648
<u>8.38.15 dirname()</u> : Renvoie le nom du dossier.....	648
<u>8.38.16 disk_free_space()</u> : Renvoie l'espace disque disponible dans le répertoire.....	649
<u>8.38.17 disk_total_space()</u> : Retourne la taille d'un dossier.....	649
<u>8.38.18 diskfree_space()</u> : Alias de <u>disk_free_space</u>	650
<u>8.38.19 fclose()</u> : Ferme un fichier.....	650
<u>8.38.20 feof()</u> : Teste la fin du fichier.....	650
<u>8.38.21 fflush()</u> : Envoie tout le contenu généré dans un fichier.....	651
<u>8.38.22 fgetc()</u> : Lit un caractère dans un fichier.....	651
<u>8.38.23 fgets()</u> : Renvoie la ligne courante et cherche les champs CSV.....	651
<u>8.38.24 fgets()</u> : Renvoie la ligne courante sur laquelle se trouve le pointeur du fichier.....	652
<u>8.38.25 fgets()</u> : Renvoie la ligne courante du fichier et élimine les balises HTML.....	653
<u>8.38.26 file_exists()</u> : Vérifie si un fichier existe.....	654
<u>8.38.27 file_get_contents()</u> : Lit tout un fichier dans une chaîne.....	654
<u>8.38.28 file_put_contents()</u> : Ecrit une chaîne dans un fichier.....	655
<u>8.38.29 file()</u> : Lit le fichier et renvoie le résultat dans un tableau.....	655
<u>8.38.30 fileatime()</u> : Renvoie la date à laquelle le fichier a été accédé pour la dernière fois.....	656
<u>8.38.31 filectime()</u> : Renvoie la date de dernier accès à un inode.....	656
<u>8.38.32 filegroup()</u> : Lire le nom du groupe.....	657
<u>8.38.33 fileinode()</u> : Renvoie le numéro d'inode du fichier.....	657
<u>8.38.34 filemtime()</u> : Renvoie la date de dernière modification du fichier.....	658
<u>8.38.35 fileowner()</u> : Renvoie le nom du propriétaire du fichier.....	658
<u>8.38.36 fileperms()</u> : Renvoie les permissions affectées à un fichier.....	658
<u>8.38.37 filesize()</u> : Renvoie la taille d'un fichier.....	660
<u>8.38.38 filetype()</u> : Retourne le type de fichier.....	660
<u>8.38.39 flock()</u> : Verrouille le fichier.....	661
<u>8.38.40 fnmatch()</u> : Repère un fichier à partir d'un masque.....	662
<u>8.38.41 fopen()</u> : Ouverture d'un fichier ou d'une URL.....	662
<u>8.38.42 fpassthru()</u> : Affiche le reste du fichier.....	665
<u>8.38.43 fputs()</u> : Formate une ligne en CSV et l'écrit dans un fichier.....	666
<u>8.38.44 fputs()</u> : Alias de <u>fwrite</u>	666
<u>8.38.45 fread()</u> : Lecture du fichier en mode binaire.....	666
<u>8.38.46 fscanf()</u> : Analyse un fichier en fonction d'un format.....	667
<u>8.38.47 fseek()</u> : Modifie la position du pointeur de fichier.....	668
<u>8.38.48 fstat()</u> : Lit les informations sur un fichier à partir d'un pointeur de fichier.....	669
<u>8.38.49 ftell()</u> : Renvoie la position du pointeur du fichier.....	670
<u>8.38.50 ftruncate()</u> : Tronque un fichier.....	670
<u>8.38.51 fwrite()</u> : Ecrit un fichier en mode binaire.....	671
<u>8.38.52 glob()</u> : Recherche des chemins qui vérifient un masque.....	671
<u>8.38.53 is_dir()</u> : Indique si le fichier est un dossier.....	672
<u>8.38.54 is_executable()</u> : Indique si le fichier est exécutable.....	673
<u>8.38.55 is_file()</u> : Indique si le fichier est un véritable fichier.....	673
<u>8.38.56 is_link()</u> : Indique si le fichier est un lien symbolique.....	674
<u>8.38.57 is_readable()</u> : Indique si un fichier est autorisé en lecture.....	674
<u>8.38.58 is_uploaded_file()</u> : Indique si le fichier a été téléchargé par HTTP POST.....	675
<u>8.38.59 is_writable()</u> : Indique si un fichier est autorisé en écriture.....	676
<u>8.38.60 is_writeable()</u> : Alias de <u>is_writable</u>	676
<u>8.38.61 link()</u> : Crée un lien.....	676
<u>8.38.62 linkinfo()</u> : Renvoie les informations d'un lien.....	677

Sommaire

8 Référence des fonctions

<u>8.38.63 lstat()</u> : Retourne les informations sur n fichier ou un lien symbolique.....	677
<u>8.38.64 mkdir()</u> : Crée un dossier.....	677
<u>8.38.65 move_uploaded_file()</u> : Déplace un fichier téléchargé.....	678
<u>8.38.66 parse_ini_file()</u> : Traite un fichier de configuration.....	679
<u>8.38.67 pathinfo()</u> : Retourne des informations sur un chemin système.....	680
<u>8.38.68 pclose()</u> : Ferme un processus de pointeur de fichier.....	681
<u>8.38.69 popen()</u> : Crée un processus de pointeur de fichier.....	681
<u>8.38.70 readfile()</u> : Affiche un fichier.....	682
<u>8.38.71 readlink()</u> : Renvoie le nom du fichier vers lequel pointe un lien symbolique.....	682
<u>8.38.72 realpath()</u> : Retourne le chemin canonique absolu.....	683
<u>8.38.73 rename()</u> : Renomme un fichier ou un dossier.....	683
<u>8.38.74 rewind()</u> : Remplace le pointeur de fichier au début.....	684
<u>8.38.75 rmdir()</u> : Efface un dossier.....	684
<u>8.38.76 set_file_buffer()</u> : Alias de stream_set_write_buffer.....	684
<u>8.38.77 stat()</u> : Renvoie les informations à propos d'un fichier.....	684
<u>8.38.78 symlink()</u> : Crée un lien symbolique.....	685
<u>8.38.79 tempnam()</u> : Crée un fichier avec un nom unique.....	686
<u>8.38.80 tmpfile()</u> : Crée un fichier temporaire.....	686
<u>8.38.81 touch()</u> : Modifie la date de modification et de dernier accès d'un fichier.....	687
<u>8.38.82 umask()</u> : Change le "umask" courant.....	687
<u>8.38.83 unlink()</u> : Efface un fichier.....	688
<u>8.39 Fonctions FriBiDi</u>	689
<u>8.39.1 Introduction</u>	689
<u>8.39.2 Pré-requis</u>	689
<u>8.39.3 Installation</u>	689
<u>8.39.4 Configuration à l'exécution</u>	689
<u>8.39.5 Types de ressources</u>	689
<u>8.39.6 Constantes pré-définies</u>	689
<u>8.40 Fonctions FTP</u>	691
<u>8.40.1 Introduction</u>	691
<u>8.40.2 Pré-requis</u>	691
<u>8.40.3 Installation</u>	691
<u>8.40.4 Configuration à l'exécution</u>	691
<u>8.40.5 Types de ressources</u>	691
<u>8.40.6 Constantes pré-définies</u>	691
<u>8.40.7 Exemples</u>	692
<u>8.40.9 ftp_cdup()</u> : Change de dossier et passe au dossier parent.....	693
<u>8.40.10 ftp_chdir()</u> : Change le dossier FTP courant.....	693
<u>8.40.11 ftp_chmod()</u> : Modifie les droits d'un fichier via FTP.....	693
<u>8.40.12 ftp_close()</u> : Ferme une connexion FTP.....	694
<u>8.40.13 ftp_connect()</u> : Ouvre une connexion FTP.....	694
<u>8.40.14 ftp_delete()</u> : Efface un fichier sur un serveur FTP.....	694
<u>8.40.15 ftp_exec()</u> : Exécute une commande sur le serveur FTP.....	694
<u>8.40.16 ftp_fget()</u> : Télécharge un fichier via FTP dans un fichier local.....	694
<u>8.40.17 ftp_fput()</u> : Charge un fichier sur un serveur FTP.....	694
<u>8.40.18 ftp_get_option()</u> : Lit différentes options pour la connexion FTP courante.....	695
<u>8.40.19 ftp_get()</u> : Télécharge un fichier depuis un serveur FTP.....	695
<u>8.40.20 ftp_login()</u> : Identification avec une connexion FTP.....	695
<u>8.40.21 ftp_mdtm()</u> : Retourne la date de dernière modification d'un fichier sur un serveur FTP.....	695

Sommaire

8 Référence des fonctions

<u>8.40.22 ftp_mkdir()</u> : Crée un dossier sur un serveur FTP.....	695
<u>8.40.23 ftp_nb_continue()</u> : Reprend le téléchargement d'un fichier (non bloquant).....	695
<u>8.40.24 ftp_nb_fget()</u> : Lit un fichier sur un serveur FTP, et l'écrit dans un fichier (non bloquant).....	696
<u>8.40.25 ftp_nb_fput()</u> : Ecrit un fichier sur un serveur FTP, et le lit depuis un fichier (non bloquant).....	696
<u>8.40.26 ftp_nb_get()</u> : Lit un fichier sur un serveur FTP, et l'écrit dans un fichier (non bloquant).....	696
<u>8.40.27 ftp_nb_put()</u> : Envoie un fichier sur le serveur FTP (non-bloquant).....	696
<u>8.40.28 ftp_nlist()</u> : Retourne la liste des fichiers dans un dossier.....	697
<u>8.40.29 ftp_pasv()</u> : Active ou désactive le mode passif.....	697
<u>8.40.30 ftp_put()</u> : Charge un fichier sur un serveur FTP.....	697
<u>8.40.31 ftp_pwd()</u> : Retourne le nom du dossier courant.....	697
<u>8.40.32 ftp_quit()</u> : Alias de ftp_close.....	697
<u>8.40.33 ftp_raw()</u> : Envoie une commande FTP brute.....	697
<u>8.40.34 ftp_rawlist()</u> : Fait une liste détaillée des fichiers dans un dossier.....	697
<u>8.40.35 ftp_rename()</u> : Renomme un fichier sur un serveur FTP.....	697
<u>8.40.36 ftp_rmdir()</u> : Efface un dossier FTP.....	698
<u>8.40.37 ftp_set_option()</u> : Modifie les options de la connexion FTP.....	698
<u>8.40.38 ftp_site()</u> : Envoie la commande SITE au serveur.....	698
<u>8.40.39 ftp_size()</u> : Retourne la taille d'un fichier.....	698
<u>8.40.40 ftp_ssl_connect()</u> : Ouvre une connexion FTP sécurisée avec SSL.....	698
<u>8.40.41 ftp_systype()</u> : Retourne un identifiant de type de serveur FTP.....	698
<u>8.41 Fonctions</u>	699
<u>8.41.1 Introduction</u>	699
<u>8.41.2 Pré-requis</u>	699
<u>8.41.3 Installation</u>	699
<u>8.41.4 Configuration à l'exécution</u>	699
<u>8.41.5 Constantes pré-définies</u>	699
<u>8.41.7 call_user_func()</u> : Appelle une fonction utilisateur.....	699
<u>8.41.8 create_function()</u> : Crée une fonction anonyme (style lambda).....	700
<u>8.41.9 func_get_arg()</u> : Retourne un élément de la liste des arguments.....	703
<u>8.41.10 func_get_args()</u> : Retourne les arguments d'une fonction sous forme de tableau.....	703
<u>8.41.11 func_num_args()</u> : Retourne le nombre d'arguments passés à la fonction.....	704
<u>8.41.12 function_exists()</u> : Indique si une fonction est définie.....	705
<u>8.41.13 get_defined_functions()</u> : Liste toutes les fonctions définies.....	705
<u>8.41.14 register_shutdown_function()</u> : Enregistre une fonction pour exécution à l'extinction.....	706
<u>8.41.15 register_tick_function()</u> : Enregistre une fonction exécutée à chaque tick.....	707
<u>8.41.16 unregister_tick_function()</u> : Annule la fonction exécutée à chaque tick.....	707
<u>8.42 Gettext (GNU)</u>	708
<u>8.42.1 Introduction</u>	708
<u>8.42.2 Pré-requis</u>	708
<u>8.42.3 Installation</u>	708
<u>8.42.4 Configuration à l'exécution</u>	708
<u>8.42.5 Types de ressources</u>	708
<u>8.42.6 Constantes pré-définies</u>	708
<u>8.42.8 bindtextdomain()</u> : Fixe le chemin d'un domaine.....	709
<u>8.42.9 dcgettext()</u> : Remplace le domaine lors d'une recherche.....	709
<u>8.42.10 dcngettext()</u> : Version plurielle de dcgettext.....	709

Sommaire

8 Référence des fonctions

<u>8.42.11 dgettext() : Remplace le domaine courant</u>	709
<u>8.42.12 dngettext() : Version plurielle de dgettext</u>	709
<u>8.42.13 gettext() : Recherche un message dans le domaine courant</u>	710
<u>8.42.14 ngettext() : Version plurielle de gettext</u>	710
<u>8.42.15 textdomain() : Fixe le domaine par défaut</u>	711
<u>8.43 Fonctions GMP</u>	712
<u>8.43.1 Introduction</u>	712
<u>8.43.2 Pré-requis</u>	712
<u>8.43.3 Installation</u>	712
<u>8.43.4 Configuration à l'exécution</u>	712
<u>8.43.5 Types de ressources</u>	712
<u>8.43.6 Constantes pré-définies</u>	713
<u>8.43.7 Exemples</u>	713
<u>8.43.8 Voir aussi</u>	713
<u>8.43.10 gmp_add() : Addition de 2 nombres GMP</u>	714
<u>8.43.11 gmp_and() : ET logique</u>	714
<u>8.43.12 gmp_clrbit() : Annule un bit</u>	715
<u>8.43.13 gmp_cmp() : Compare des nombres GMP</u>	715
<u>8.43.14 gmp_com() : Calcule le complémentaire d'un nombre</u>	716
<u>8.43.15 gmp_div_q() : Divisions de 2 nombres GMP</u>	716
<u>8.43.16 gmp_div_qr() : Divise deux nombres GMP</u>	717
<u>8.43.17 gmp_div_r() : Reste de la division de deux nombres GMP</u>	717
<u>8.43.18 gmp_div() : Alias de gmp_div_q</u>	717
<u>8.43.19 gmp_divexact() : Division exacte de nombres GMP</u>	718
<u>8.43.20 gmp_fact() : Factorielle GMP</u>	718
<u>8.43.21 gmp_gcd() : PGCD</u>	718
<u>8.43.22 gmp_gcdext() : PGCD étendu</u>	719
<u>8.43.23 gmp_hamdist() : Distance de Hamming</u>	719
<u>8.43.24 gmp_init() : Crée un nombre GMP</u>	720
<u>8.43.25 gmp_intval() : Convertit un nombre GMP en entier</u>	720
<u>8.43.26 gmp_invert() : Inverse modulo</u>	721
<u>8.43.27 gmp_jacobi() : Symbole de Jacobi</u>	721
<u>8.43.28 gmp_legendre() : Symbole de Legendre</u>	721
<u>8.43.29 gmp_mod() : Modulo GMP</u>	722
<u>8.43.30 gmp_mul() : Multiplication de 2 nombres GMP</u>	722
<u>8.43.31 gmp_neg() : Opposé de nombre GMP</u>	722
<u>8.43.32 gmp_or() : OU logique</u>	723
<u>8.43.33 gmp_perfect_square() : Carré parfait GMP</u>	723
<u>8.43.34 gmp_popcount() : Compte de population</u>	724
<u>8.43.35 gmp_pow() : Puissance</u>	724
<u>8.43.36 gmp_powm() : Puissance et modulo</u>	724
<u>8.43.37 gmp_prob_prime() : Nombre GMP probablement premier</u>	725
<u>8.43.38 gmp_random() : Nombre GMP aléatoire</u>	725
<u>8.43.39 gmp_scan0() : Recherche 0</u>	726
<u>8.43.40 gmp_scan1() : Recherche 1</u>	726
<u>8.43.41 gmp_setbit() : Modifie un bit</u>	727
<u>8.43.42 gmp_sign() : Signe du nombre GMP</u>	727
<u>8.43.43 gmp_sqrt() : Racine carrée GMP</u>	728
<u>8.43.44 gmp_sqrtrem() : Racine carrée avec reste GMP</u>	728
<u>8.43.45 gmp_strval() : Convertit un nombre GMP en chaîne</u>	728
<u>8.43.46 gmp_sub() : Soustraction de 2 nombres GMP</u>	729

Sommaire

8 Référence des fonctions

8.43.47 <u>gmp_xor()</u> : OU exclusif logique.....	729
8.44 HTTP.....	730
8.44.1 Introduction.....	730
8.44.2 Pré-requis.....	730
8.44.3 Installation.....	730
8.44.4 Configuration à l'exécution.....	730
8.44.5 Types de ressources.....	730
8.44.6 Constantes pré-définies.....	730
8.44.8 <u>headers_list()</u> : Retourne la liste des en-têtes de réponse du script courant.....	730
8.44.9 <u>headers_sent()</u> : Indique si les en-têtes HTTP ont déjà été envoyés.....	731
8.44.10 <u>setcookie()</u> : Envoie un cookie.....	732
8.44.11 <u>setrawcookie()</u> : Envoie un cookie sans encoder sa valeur en URL.....	735
8.45 Hyperwave.....	736
8.45.1 Introduction.....	736
8.45.2 Pré-requis.....	738
8.45.3 Installation.....	738
8.45.4 Intégration avec Apache.....	738
8.45.5 Configuration à l'exécution.....	739
8.45.6 Types de ressources.....	739
8.45.7 Constantes pré-définies.....	740
8.45.8 A faire.....	740
8.45.10 <u>hw_changeobject()</u> : Change les attributs d'un objet (obsolète).....	741
8.45.11 <u>hw_Children()</u> : Liste des objets ids des objets fils.....	741
8.45.12 <u>hw_ChildrenObj()</u> : Liste des objets records des objets fils.....	742
8.45.13 <u>hw_Close()</u> : Ferme la connexion Hyperwave.....	742
8.45.14 <u>hw_Connect()</u> : Ouvre une connexion Hyperwave.....	742
8.45.15 <u>hw_connection_info()</u> : Affiche des informations sur la connexion au serveur Hyperwave.....	742
8.45.16 <u>hw_cp()</u> : Copie des objets HyperWave.....	742
8.45.17 <u>hw_Deleteobject()</u> : Efface des objets.....	743
8.45.18 <u>hw_DocByAnchor()</u> : Identifiant d'objet de l'objet dans l'ancrage.....	743
8.45.19 <u>hw_DocByAnchorObj()</u> : Attributs de l'objet dans l'ancrage.....	743
8.45.20 <u>hw_Document_Attributes()</u> : Object record de hw_document.....	743
8.45.21 <u>hw_Document_BodyTag()</u> : Balise de corps d'un document.....	743
8.45.22 <u>hw_Document_Content()</u> : Contenu d'un document.....	744
8.45.23 <u>hw_Document_SetContent()</u> : Modifie/remplace le contenu d'un document.....	744
8.45.24 <u>hw_Document_Size()</u> : Taille d'un document.....	744
8.45.25 <u>hw_dummy()</u> : Fonction sans objet pour Hyperwave.....	744
8.45.26 <u>hw_EditText()</u> : Retourne un document texte.....	744
8.45.27 <u>hw_Error()</u> : Retourne le code d'erreur.....	745
8.45.28 <u>hw_ErrorMsg()</u> : Retourne un message d'erreur.....	745
8.45.29 <u>hw_Free_Document()</u> : Détruit un document.....	745
8.45.30 <u>hw_GetAnchors()</u> : Identifiants des ancrages d'un document.....	745
8.45.31 <u>hw_GetAnchorsObj()</u> : Attributs des ancrages d'un document.....	745
8.45.32 <u>hw_GetAndLock()</u> : Retourne les attributs et verrouille l'objet.....	745
8.45.33 <u>hw_GetChildColl()</u> : Identifiant d'objets des groupes fils.....	746
8.45.34 <u>hw_GetChildCollObj()</u> : Objet record d'un groupe d'enfants.....	746
8.45.35 <u>hw_GetChildDocColl()</u> : Identifiant des documents fils d'un groupe.....	746
8.45.36 <u>hw_GetChildDocCollObj()</u> : Attributs des documents fils d'un groupe.....	746
8.45.37 <u>hw_GetObject()</u> : Lit les attributs d'un objet Hyperwave.....	746
8.45.38 <u>hw_GetObjectByQuery()</u> : Recherche un objet.....	747

Sommaire

8 Référence des fonctions

<u>8.45.39 hw</u> <u>GetObjectByQueryColl()</u> : Recherche un objet dans un groupe.....	747
<u>8.45.40 hw</u> <u>GetObjectByQueryCollObj()</u> : Recherche un objet dans un groupe.....	748
<u>8.45.41 hw</u> <u>GetObjectByQueryObj()</u> : Recherche un objet.....	748
<u>8.45.42 hw</u> <u>GetParents()</u> : Identifiant d'objet des parents.....	748
<u>8.45.43 hw</u> <u>GetParentsObj()</u> : Attributs des parents.....	748
<u>8.45.44 hw</u> <u>getrellink()</u> : Crée le lien depuis la source vers le destinataire, relativement à rootid.....	748
<u>8.45.45 hw</u> <u>GetRemote()</u> : Retourne un document distant.....	749
<u>8.45.46 hw</u> <u>getremotechildren()</u> : Retourne les fils d'un document distant.....	749
<u>8.45.47 hw</u> <u>GetSrcByDestObj()</u> : Retourne les ancrages qui pointent sur un objet.....	749
<u>8.45.48 hw</u> <u>GetText()</u> : Retourne un document texte.....	749
<u>8.45.49 hw</u> <u>getusername()</u> : Nom de l'utilisateur actuellement identifié.....	750
<u>8.45.50 hw</u> <u>Identify()</u> : Identifie un utilisateur.....	750
<u>8.45.51 hw</u> <u>InCollections()</u> : Vérifie qu'un identifiant d'objet est dans un groupe.....	750
<u>8.45.52 hw</u> <u>Info()</u> : Informations à propos d'une connexion.....	751
<u>8.45.53 hw</u> <u>InsColl()</u> : Insère un groupe.....	751
<u>8.45.54 hw</u> <u>InsDoc()</u> : Insère un document.....	751
<u>8.45.55 hw</u> <u>insertanchors()</u> : Insère uniquement des ancrages dans du texte.....	751
<u>8.45.56 hw</u> <u>InsertDocument()</u> : Insère un document dans un groupe.....	751
<u>8.45.57 hw</u> <u>InsertObject()</u> : Insère un objet record.....	752
<u>8.45.58 hw</u> <u>mapid()</u> : Représente un id global en un id virtuel local.....	752
<u>8.45.59 hw</u> <u>Modifyobject()</u> : Modifie les attributs d'objet record.....	752
<u>8.45.60 hw</u> <u>mv()</u> : Déplace un objet.....	754
<u>8.45.61 hw</u> <u>New Document()</u> : Crée un nouveau document.....	754
<u>8.45.62 hw</u> <u>objrec2array()</u> : Convertit les attributs d'un objet en tableau.....	754
<u>8.45.63 hw</u> <u>Output Document()</u> : Affiche hw document.....	755
<u>8.45.64 hw</u> <u>pConnect()</u> : Crée une connexion persistante.....	755
<u>8.45.65 hw</u> <u>PipeDocument()</u> : Retourne un document.....	755
<u>8.45.66 hw</u> <u>Root()</u> : Objet id de la racine.....	755
<u>8.45.67 hw</u> <u>setlinkroot()</u> : Configure l'identifiant vers lequel les liens sont calculés.....	756
<u>8.45.68 hw</u> <u>stat()</u> : Retourne le statut de la chaîne.....	756
<u>8.45.69 hw</u> <u>Unlock()</u> : Déverrouille un objet.....	756
<u>8.45.70 hw</u> <u>Who()</u> : Liste des utilisateurs actuellement identifiés.....	756
<u>8.46 Hyperwave API</u>	757
<u>8.46.1 Introduction</u>	757
<u>8.46.2 Pré-requis</u>	757
<u>8.46.3 Installation</u>	757
<u>8.46.4 intégration dans Apache</u>	757
<u>8.46.5 Configuration à l'exécution</u>	757
<u>8.46.6 Types de ressources</u>	758
<u>8.46.7 Constantes pré-définies</u>	758
<u>8.46.8 Classes</u>	758
<u>8.46.10 hw</u> <u>api attribute->langdepvalue()</u> : Retourne la valeur d'un attribut dans une langue.....	760
<u>8.46.11 hw</u> <u>api attribute->value()</u> : Retourne la valeur d'un attribut.....	760
<u>8.46.12 hw</u> <u>api attribute->values()</u> : Retourne toutes les valeurs de l'attribut.....	760
<u>8.46.13 hw</u> <u>api attribute()</u> : Crée une nouvelle instance de la classe <u>hw api attribute</u>	760
<u>8.46.14 hw</u> <u>api->checkin()</u> : Archive un objet.....	760
<u>8.46.15 hw</u> <u>api->checkout()</u> : Extrait un objet.....	761
<u>8.46.16 hw</u> <u>api->children()</u> : Retourne le fils d'un objet.....	761

Sommaire

8 Référence des fonctions

<u>8.46.17 hw_api_content->mimetype()</u> : Retourne le type MIME.....	762
<u>8.46.18 hw_api_content->read()</u> : Lit le contenu.....	762
<u>8.46.19 hw_api->content()</u> : Retourne le contenu d'un objet.....	762
<u>8.46.20 hw_api->copy()</u> : Copie physique.....	762
<u>8.46.21 hw_api->dbstat()</u> : Retourne des statistiques sur le serveur de bases.....	762
<u>8.46.22 hw_api->dcstat()</u> : Retourne des statistiques sur le serveur de cache de document.....	762
<u>8.46.23 hw_api->dstanchors()</u> : Liste toutes les ancrs de destination.....	763
<u>8.46.24 hw_api->dstofsrcanchor()</u> : Retourne la destination d'une ancre source.....	763
<u>8.46.25 hw_api_error->count()</u> : Retourne le nombre de raisons d'erreurs.....	763
<u>8.46.26 hw_api_error->reason()</u> : Retourne la première raison d'erreur.....	763
<u>8.46.27 hw_api->find()</u> : Recherche des objets.....	763
<u>8.46.28 hw_api->ftstat()</u> : Retourne des statistiques sur le serveur de textes.....	764
<u>8.46.29 hwapi_hgcsp()</u> : Retourne un objet de classe hw_api.....	764
<u>8.46.30 hw_api->hwstat()</u> : Retourne les statistiques sur le serveur Hyperwave.....	764
<u>8.46.31 hw_api->identify()</u> : S'identifie auprès du serveur Hyperwave.....	764
<u>8.46.32 hw_api->info()</u> : Retourne des informations sur la configuration du serveur.....	765
<u>8.46.33 hw_api->insert()</u> : Insère un nouvel objet.....	765
<u>8.46.34 hw_api->insertanchor()</u> : Insère un nouvel objet de type ancre.....	765
<u>8.46.35 hw_api->insertcollection()</u> : Insère un nouvel objet de type collection.....	765
<u>8.46.36 hw_api->insertdocument()</u> : Insère un nouvel objet de type document.....	766
<u>8.46.37 hw_api->link()</u> : Crée un lien avec un objet.....	766
<u>8.46.38 hw_api->lock()</u> : Verrouille un objet.....	766
<u>8.46.39 hw_api->move()</u> : Déplace un objet d'une collection à l'autre.....	767
<u>8.46.40 hw_api_content()</u> : Crée une nouvelle instance de la classe hw_api_content.....	767
<u>8.46.41 hw_api_object->assign()</u> : Clone un objet.....	767
<u>8.46.42 hw_api_object->attributable()</u> : Vérifie si un attribut est éditable.....	767
<u>8.46.43 hw_api_object->count()</u> : Retourne le nombre d'attributs.....	767
<u>8.46.44 hw_api_object->insert()</u> : Insère un nouvel attribut HyperWave.....	767
<u>8.46.45 hw_api_object()</u> : Crée une nouvelle instance de la classe hw_api_object.....	767
<u>8.46.46 hw_api_object->remove()</u> : Supprime un attribut HyperWave.....	768
<u>8.46.47 hw_api_object->title()</u> : Retourne le titre de l'attribut.....	768
<u>8.46.48 hw_api_object->value()</u> : Retourne la valeur de l'attribut.....	768
<u>8.46.49 hw_api->object()</u> : Lit les informations d'attribut.....	768
<u>8.46.50 hw_api->objectbyanchor()</u> : Retourne l'objet propriétaire d'une ancre.....	769
<u>8.46.51 hw_api->parents()</u> : Retourne le parent d'un objet.....	769
<u>8.46.52 hw_api_reason->description()</u> : Retourne la description d'une raison.....	770
<u>8.46.53 hw_api_reason->type()</u> : Retourne le type de raison.....	770
<u>8.46.54 hw_api->remove()</u> : Efface un objet.....	770
<u>8.46.55 hw_api->replace()</u> : Remplace un objet.....	770
<u>8.46.56 hw_api->setcommittedversion()</u> : Valide une version autre que la dernière.....	771
<u>8.46.57 hw_api->srcanchors()</u> : Retourne la liste de toutes les ancrs.....	771
<u>8.46.58 hw_api->srcsofdst()</u> : Retourne les sources d'un objet de destination.....	771
<u>8.46.59 hw_api->unlock()</u> : Déverrouille un objet HyperWave verrouillé.....	771
<u>8.46.60 hw_api->user()</u> : Retourne le propriétaire d'un objet.....	772
<u>8.46.61 hw_api->userlist()</u> : Retourne la liste des utilisateurs connectés.....	772
<u>8.47 Fonctions Firebird/InterBase.....</u>	773
<u>8.47.1 Introduction.....</u>	773
<u>8.47.2 Pré-requis.....</u>	773
<u>8.47.3 Installation.....</u>	773

Sommaire

8 Référence des fonctions

<u>8.47.4 Configuration à l'exécution</u>	773
<u>8.47.5 Types de ressources</u>	774
<u>8.47.6 Constantes pré-définies</u>	775
<u>8.47.8 <code>ibase_affected_rows()</code> : Retourne le nombre de lignes affectées par la dernière requête iBase</u>	778
<u>8.47.9 <code>ibase_backup()</code> : Effectue une sauvegarde de base de données InterBase</u>	778
<u>8.47.10 <code>ibase_blob_add()</code> : Ajoute des données dans un BLOB iBase fraîchement créé</u>	778
<u>8.47.11 <code>ibase_blob_cancel()</code> : Annule la création d'un BLOB iBase</u>	778
<u>8.47.12 <code>ibase_blob_close()</code> : Ferme un BLOB iBase</u>	779
<u>8.47.13 <code>ibase_blob_create()</code> : Crée un BLOB iBase pour ajouter des données</u>	779
<u>8.47.14 <code>ibase_blob_echo()</code> : Affiche le contenu d'un BLOB iBase au navigateur</u>	779
<u>8.47.15 <code>ibase_blob_get()</code> : Lit len octets de données dans un BLOB iBase ouvert</u>	779
<u>8.47.16 <code>ibase_blob_import()</code> : Crée un BLOB iBase, y copie un fichier et le referme</u>	780
<u>8.47.17 <code>ibase_blob_info()</code> : Retourne la taille d'un BLOB iBase et d'autres informations utiles</u>	780
<u>8.47.18 <code>ibase_blob_open()</code> : Ouvre un BLOB iBase pour récupérer des parties de données</u>	781
<u>8.47.19 <code>ibase_close()</code> : Ferme une connexion à une base de données Interbase</u>	781
<u>8.47.20 <code>ibase_commit_ret()</code> : Valide une transaction iBase sans la refermer</u>	781
<u>8.47.21 <code>ibase_commit()</code> : Valide une transaction iBase</u>	781
<u>8.47.22 <code>ibase_connect()</code> : Ouvre une connexion à une base de données InterBase</u>	782
<u>8.47.23 <code>ibase_db_info()</code> : Demande des statistiques sur une base de données Interbase</u>	782
<u>8.47.24 <code>ibase_delete_user()</code> : Efface un utilisateur d'une base de données de sécurité (uniquement pour IB6 ou plus récent)</u>	783
<u>8.47.25 <code>ibase_drop_db()</code> : Supprime une base de données iBase</u>	783
<u>8.47.26 <code>ibase_errcode()</code> : Retourne le code d'erreur iBase</u>	783
<u>8.47.27 <code>ibase_errmsg()</code> : Retourne un message d'erreur</u>	783
<u>8.47.28 <code>ibase_execute()</code> : Exécute une requête iBase préparée</u>	783
<u>8.47.29 <code>ibase_fetch_assoc()</code> : Récupère une ligne du résultat d'une requête dans un tableau associatif</u>	784
<u>8.47.30 <code>ibase_fetch_object()</code> : Lit une ligne dans une base Interbase dans un objet</u> ..	784
<u>8.47.31 <code>ibase_fetch_row()</code> : Lit une ligne dans une base Interbase</u>	785
<u>8.47.32 <code>ibase_field_info()</code> : Lit les informations sur un champ iBase</u>	785
<u>8.47.33 <code>ibase_free_event_handler()</code> : Libère un gestionnaire d'événements iBase</u>	786
<u>8.47.34 <code>ibase_free_query()</code> : Libère la mémoire réservée par une requête préparée</u>	786
<u>8.47.35 <code>ibase_free_result()</code> : Libère un résultat iBase</u>	786
<u>8.47.36 <code>ibase_gen_id()</code> : Incrmente le générateur donné et retourne sa nouvelle valeur</u>	786
<u>8.47.37 <code>ibase_maintain_db()</code> : Exécute une commande de maintenance sur une base de données Interbase</u>	787
<u>8.47.38 <code>ibase_modify_user()</code> : Modifie un utilisateur dans une base de données de sécurité (uniquement pour InterBase6 ou plus récent)</u>	787
<u>8.47.39 <code>ibase_name_result()</code> : Assigne un nom à un jeu de résultats iBase</u>	787
<u>8.47.40 <code>ibase_num_fields()</code> : Retourne le nombre de colonnes dans un résultat iBase</u>	787
<u>8.47.41 <code>ibase_num_params()</code> : Retourne le nombre de paramètres dans une</u>	

Sommaire

8 Référence des fonctions

<u>requête préparée iBase</u>	788
<u>8.47.42 ibase_param_info() : Retourne des informations à propos d'un paramètre dans une requête préparée iBase</u>	788
<u>8.47.43 ibase_pconnect() : Ouvre une connexion persistante à une base de données InterBase</u>	788
<u>8.47.44 ibase_prepare() : Prépare une requête iBase pour lier les paramètres et l'exécuter ultérieurement</u>	789
<u>8.47.45 ibase_query() : Exécute une requête sur une base iBase</u>	789
<u>8.47.46 ibase_restore() : Restaure une sauvegarde de base de données Interbase</u>	790
<u>8.47.47 ibase_rollback_ret() : Annule une transaction sans la refermer</u>	790
<u>8.47.48 ibase_rollback() : Annule une transaction iBase</u>	790
<u>8.47.49 ibase_server_info() : Demande des informations sur le serveur Interbase</u>	790
<u>8.47.50 ibase_service_attach() : Connexion au service de gestion Interbase</u>	790
<u>8.47.51 ibase_service_detach() : Déconnexion du service de gestion Interbase</u>	791
<u>8.47.52 ibase_set_event_handler() : Enregistre une fonction de callback sur un événement iBase</u>	791
<u>8.47.53 ibase_timefmt() : Fixe le format de date pour les prochaines requêtes</u>	791
<u>8.47.54 ibase_trans() : Prépare une transaction iBase</u>	792
<u>8.47.55 ibase_wait_event() : Attend un événement iBase</u>	792
<u>8.48 ICAP</u>	793
<u>8.48.1 Introduction</u>	793
<u>8.48.2 Pré-requis</u>	793
<u>8.48.3 Installation</u>	793
<u>8.48.4 Configuration à l'exécution</u>	793
<u>8.48.5 Types de ressources</u>	793
<u>8.48.6 Constantes pré-définies</u>	793
<u>8.48.8 icap_create_calendar() : Crée un nouveau calendrier</u>	794
<u>8.48.9 icap_delete_calendar() : Supprime un calendrier</u>	794
<u>8.48.10 icap_delete_event() : Efface un événement dans un agenda ICAP</u>	794
<u>8.48.11 icap_fetch_event() : Recherche un événement dans le calendrier</u>	794
<u>8.48.12 icap_list_alarms() : Retourne une liste d'événements qui ont une alarme prévue à une date</u>	795
<u>8.48.13 icap_list_events() : Retourne une liste d'événements entre deux dates</u>	795
<u>8.48.14 icap_open() : Ouvre une connexion ICAP</u>	795
<u>8.48.15 icap_rename_calendar() : Renomme un calendrier</u>	795
<u>8.48.16 icap_reopen() : Réouvre un flux ICAP pour un nouveau calendrier</u>	796
<u>8.48.17 icap_snooze() : Eteind l'alarme d'un événement</u>	796
<u>8.48.18 icap_store_event() : Enregistre un événement dans un agenda ICAP</u>	796
<u>8.49 Iconv</u>	797
<u>8.49.1 Introduction</u>	797
<u>8.49.2 Pré-requis</u>	797
<u>8.49.3 Installation</u>	797
<u>8.49.4 Configuration à l'exécution</u>	797
<u>8.49.5 Types de ressources</u>	798
<u>8.49.6 Constantes pré-définies</u>	798
<u>8.49.7 Voir aussi</u>	798
<u>8.49.9 iconv_mime_decode_headers() : Décode des en-têtes MIME multiples</u>	799
<u>8.49.10 iconv_mime_decode() : Décode un champ d'en-tête MIME</u>	800
<u>8.49.11 iconv_mime_encode() : Construit un en-tête MIME avec les champs field_name et field_value</u>	801

Sommaire

8 Référence des fonctions

<u>8.49.12 iconv_set_encoding()</u> : Modifie le jeu courant de caractères d'encodage.....	802
<u>8.49.13 iconv_strlen()</u> : Retourne le nombre de caractères d'une chaîne.....	802
<u>8.49.14 iconv_strpos()</u> : Trouve la position de la première occurrence de needle dans haystack.....	803
<u>8.49.15 iconv_strrpos()</u> : Trouve la position de la dernière occurrence d'un élément dans une chaîne, à partir d'un offset.....	803
<u>8.49.16 iconv_substr()</u> : Retourne une partie de chaîne iconv.....	804
<u>8.49.17 iconv()</u> : Convertit une chaîne dans un jeu de caractères.....	804
<u>8.49.18 ob_iconv_handler()</u> : Gestionnaire de sortie pour maîtriser le jeu de caractères de sortie.....	804
<u>8.50 Fonctions ID3</u>	806
<u>8.50.1 Introduction</u>	806
<u>8.50.2 Pré-requis</u>	806
<u>8.50.3 Installation</u>	806
<u>8.50.4 Configuration à l'exécution</u>	806
<u>8.50.5 Types de ressources</u>	806
<u>8.50.6 Constantes pré-définies</u>	806
<u>8.50.8 id3_get_frame_short_name()</u> : Récupère le nom court d'un champs ID3v2....	807
<u>8.50.9 id3_get_genre_id()</u> : Récupération d'un id pour un genre.....	807
<u>8.50.10 id3_get_genre_list()</u> : Récupère toutes les valeurs possibles du genre.....	808
<u>8.50.11 id3_get_genre_name()</u> : Récupère le nom pour un id de genre.....	811
<u>8.50.12 id3_get_tag()</u> : Récupère toutes les informations stockées dans un tag ID3..	812
<u>8.50.13 id3_get_version()</u> : Récupère la version d'un tag ID3.....	813
<u>8.50.14 id3_remove_tag()</u> : Efface un tag ID3.....	813
<u>8.50.15 id3_set_tag()</u> : Met à jour les informations stockées dans un tag ID3.....	814
<u>8.51 Informix</u>	816
<u>8.51.1 Introduction</u>	816
<u>8.51.2 Pré-requis</u>	816
<u>8.51.3 Installation</u>	816
<u>8.51.4 Configuration à l'exécution</u>	816
<u>8.51.5 Types de ressources</u>	819
<u>8.51.6 Constantes pré-définies</u>	819
<u>8.51.8 ifx_blobinfile_mode()</u> : Choisit le mode par défaut des objets BLOB pour toutes les requêtes SELECT.....	820
<u>8.51.9 ifx_byteasvarchar()</u> : Choisit le mode par défaut des objets BYTE.....	820
<u>8.51.10 ifx_close()</u> : Ferme une connexion à un serveur Informix.....	820
<u>8.51.11 ifx_connect()</u> : Ouvre une connexion à un serveur Informix.....	821
<u>8.51.12 ifx_copy_blob()</u> : Duplique un objet BLOB.....	821
<u>8.51.13 ifx_create_blob()</u> : Crée un objet BLOB.....	821
<u>8.51.14 ifx_create_char()</u> : Crée un objet char.....	822
<u>8.51.15 ifx_do()</u> : Exécute une requête SQL préparée.....	822
<u>8.51.16 ifx_error()</u> : Retourne le code d'erreur de la dernière requête Informix.....	822
<u>8.51.17 ifx_errormsg()</u> : Retourne le message d'erreur de la dernière requête Informix.....	823
<u>8.51.18 ifx_fetch_row()</u> : Retourne une ligne sous la forme d'un tableau énuméré....	823
<u>8.51.19 ifx_fieldproperties()</u> : Liste les propriétés des champs SQL.....	824
<u>8.51.20 ifx_fieldtypes()</u> : Liste les champs Informix SQL.....	824
<u>8.51.21 ifx_free_blob()</u> : Supprime un objet BLOB.....	825
<u>8.51.22 ifx_free_char()</u> : Supprime un objet char.....	825
<u>8.51.23 ifx_free_result()</u> : Libère les ressources prises par un résultat.....	825
<u>8.51.24 ifx_get_blob()</u> : Retourne le contenu d'un objet BLOB.....	825

Sommaire

8 Référence des fonctions

<u>8.51.25 ifx_get_char()</u> : Retourne le contenu d'un objet char.....	825
<u>8.51.26 ifx_getsqlca()</u> : Retourne le contenu de la variable sqlca.sqlerrd[0..5] après une requête.....	825
<u>8.51.27 ifx_htmltbl_result()</u> : Lit toutes les lignes d'un tableau, et le met sous la forme d'un tableau HTML.....	826
<u>8.51.28 ifx_nullformat()</u> : Modifie le mode par défaut de lecture des valeurs.....	826
<u>8.51.29 ifx_num_fields()</u> : Retourne le nombre de colonnes dans une requête.....	827
<u>8.51.30 ifx_num_rows()</u> : Compte le nombre de lignes déjà lues dans un résultat....	827
<u>8.51.31 ifx_pconnect()</u> : Ouvre une connexion persistante à un serveur Informix.....	827
<u>8.51.32 ifx_prepare()</u> : Prépare une requête SQL pour l'exécution.....	828
<u>8.51.33 ifx_query()</u> : Envoie une requête Informix.....	828
<u>8.51.34 ifx_textasvarchar()</u> : Choisit le mode par défaut des objets text.....	829
<u>8.51.35 ifx_update_blob()</u> : Modifie le contenu d'un objet BLOB.....	829
<u>8.51.36 ifx_update_char()</u> : Modifie le contenu d'un objet char.....	830
<u>8.51.37 ifxus_close_slob()</u> : Ferme un objet SLOB.....	830
<u>8.51.38 ifxus_create_slob()</u> : Crée un objet SLOB et l'ouvre.....	830
<u>8.51.39 ifxus_free_slob()</u> : Supprime un objet SLOB.....	830
<u>8.51.40 ifxus_open_slob()</u> : Ouvre un objet SLOB.....	830
<u>8.51.41 ifxus_read_slob()</u> : Lit n bytes d'un objet SLOB.....	831
<u>8.51.42 ifxus_seek_slob()</u> : Fixe le fichier courant ou la position courante.....	831
<u>8.51.43 ifxus_tell_slob()</u> : Retourne le fichier courant ou la position courante.....	831
<u>8.51.44 ifxus_write_slob()</u> : Ecrit une chaîne dans un objet SLOB.....	831
<u>8.52 Fonctions d'administration d'IIS</u>	832
<u>8.52.1 Introduction</u>	832
<u>8.52.2 Pré-requis</u>	832
<u>8.52.3 Installation</u>	832
<u>8.52.4 Configuration à l'exécution</u>	832
<u>8.52.5 Types de ressources</u>	832
<u>8.52.6 Constantes pré-définies</u>	832
<u>8.52.8 iis_get_dir_security()</u> : Lit la configuration de sécurité du dossier.....	833
<u>8.52.9 iis_get_script_map()</u> : Lit le mappage de script dans un dossier virtuel donné.....	833
<u>8.52.10 iis_get_server_by_comment()</u> : Retourne le numéro d'instance associé à un Comment.....	833
<u>8.52.11 iis_get_server_by_path()</u> : Retourne le numéro d'instance associé avec le Path.....	833
<u>8.52.12 iis_get_server_rights()</u> : Lit les droits du serveur.....	833
<u>8.52.13 iis_get_service_state()</u> : Retourne l'état du service défini par ServiceId.....	833
<u>8.52.14 iis_remove_server()</u> : Supprime le serveur IIS représenté par ServerInstance.....	833
<u>8.52.15 iis_set_app_settings()</u> : Crée un espace d'exécution pour une application dans un dossier virtuel.....	834
<u>8.52.16 iis_set_dir_security()</u> : Modifie la configuration de sécurité du dossier.....	834
<u>8.52.17 iis_set_script_map()</u> : Modifie le mappage de script pour un dossier virtuel.....	834
<u>8.52.18 iis_set_server_rights()</u> : Modifie les droits du serveur.....	834
<u>8.52.19 iis_start_server()</u> : Démarre le serveur web virtuel.....	834
<u>8.52.20 iis_start_service()</u> : Démarre le service IIS identifié par ServiceId.....	834
<u>8.52.21 iis_stop_server()</u> : Stoppe le serveur web virtuel.....	834
<u>8.52.22 iis_stop_service()</u> : Stoppe le service IIS identifié par ServiceId.....	834
<u>8.53 Images</u>	835
<u>8.53.1 Introduction</u>	835

Sommaire

8 Référence des fonctions

<u>8.53.2 Pré-requis</u>	835
<u>8.53.3 Installation</u>	836
<u>8.53.4 Configuration à l'exécution</u>	837
<u>8.53.5 Types de ressources</u>	837
<u>8.53.6 Constantes pré-définies</u>	837
<u>8.53.7 Exemples</u>	838
<u>8.53.9 getimagesize() : Retourne la taille d'une image</u>	840
<u>8.53.10 image_type_to_extension() : Retourne l'extension du fichier pour le type d'image</u>	842
<u>8.53.11 image_type_to_mime_type() : Lit le Mime-Type d'un type d'image</u>	842
<u>8.53.12 image2wbmp() : Crée une image WBMP</u>	843
<u>8.53.13 imagealphablending() : Modifie le mode de blending d'une image</u>	843
<u>8.53.14 imageantialias() : Utiliser ou non les fonctions d'antialias</u>	844
<u>8.53.15 imagearc() : Dessine une ellipse partielle</u>	844
<u>8.53.16 imagechar() : Dessine un caractère horizontalement</u>	845
<u>8.53.17 imagecharup() : Dessine un caractère verticalement</u>	845
<u>8.53.18 imagecolorallocate() : Alloue une couleur pour une image</u>	846
<u>8.53.19 imagecolorallocatealpha() : Alloue une couleur à une image</u>	846
<u>8.53.20 imagecolorat() : Retourne l'index de la couleur d'un pixel donné</u>	847
<u>8.53.21 imagecolorclosest() : Retourne l'index de la couleur la plus proche d'une couleur donnée</u>	847
<u>8.53.22 imagecolorclosestalpha() : Retourne la couleur la plus proche, en tenant compte du canal alpha</u>	848
<u>8.53.23 imagecolorclosesthw() : Lit l'index de la couleur la plus proche d'une couleur spécifiée avec sa teinte, blanc et noir</u>	848
<u>8.53.24 imagecolordeallocate() : Supprime une couleur d'une image</u>	848
<u>8.53.25 imagecolorexact() : Retourne l'index de la couleur donnée</u>	848
<u>8.53.26 imagecolorexactalpha() : Retourne l'index d'une couleur avec son canal alpha</u>	849
<u>8.53.27 imagecolormatch() : Fait correspondre un peu plus les couleurs de la version palette d'une image aux couleurs de sa version truecolor</u>	849
<u>8.53.28 imagecolorresolve() : Retourne l'index de la couleur donnée, ou la plus proche possible</u>	849
<u>8.53.29 imagecolorresolvealpha() : Retourne un index de couleur ou son alternative la plus proche, y compris le canal alpha</u>	850
<u>8.53.30 imagecolorset() : Change la couleur dans une palette à l'index donné</u>	850
<u>8.53.31 imagecolorsforindex() : Retourne la couleur associée à un index</u>	850
<u>8.53.32 imagecolorstotal() : Calcule le nombre de couleurs d'une palette</u>	851
<u>8.53.33 imagecolortransparent() : Définit la couleur transparente</u>	851
<u>8.53.34 imageconvolution() : Applique une matrice de la convolution 3x3, en utilisant la division de coefficient et l'excentrage</u>	851
<u>8.53.35 imagecopy() : Copie une partie d'une image</u>	852
<u>8.53.36 imagecopymerge() : Copie et fusionne une partie d'une image</u>	852
<u>8.53.37 imagecopymergegray() : Copie et fusionne une partie d'une image en niveaux de gris</u>	852
<u>8.53.38 imagecopyresampled() : Copie, redimensionne, rééchantillonne une image</u>	853
<u>8.53.39 imagecopyresized() : Copie et redimensionne une partie d'une image</u>	853
<u>8.53.40 imagecreate() : Crée une nouvelle image à palette</u>	854
<u>8.53.41 imagecreatefromgd() : Crée une nouvelle image à partir d'un fichier GD ou d'une URL</u>	854

Sommaire

8 Référence des fonctions

<u>8.53.42 imagecreatefromgd2() : Crée une nouvelle image à partir d'un fichier GD2 ou d'une URL.....</u>	854
<u>8.53.43 imagecreatefromgd2part() : Crée une nouvelle image à partir d'une partie de fichier GD2 ou d'une URL.....</u>	855
<u>8.53.44 imagecreatefromgif() : Crée une nouvelle image à partir d'un fichier ou d'une URL.....</u>	855
<u>8.53.45 imagecreatefromjpeg() : Crée une nouvelle image JPEG à partir d'un fichier ou d'une URL.....</u>	856
<u>8.53.46 imagecreatefrompng() : Crée une nouvelle image PNG à partir d'un fichier ou d'une URL.....</u>	856
<u>8.53.47 imagecreatefromstring() : Crée une image à partir d'une chaîne.....</u>	857
<u>8.53.48 imagecreatefromwbmp() : Crée une nouvelle image WBMP à partir d'un fichier ou d'une URL.....</u>	857
<u>8.53.49 imagecreatefromxbm() : Crée une nouvelle image XBM à partir d'un fichier ou d'une URL.....</u>	858
<u>8.53.50 imagecreatefromxpm() : Crée une nouvelle image XPM à partir d'un fichier ou d'une URL.....</u>	858
<u>8.53.51 imagecreatetruecolor() : Crée une nouvelle image en vraies couleurs.....</u>	858
<u>8.53.52 imagedashedline() : Dessine une ligne pointillée.....</u>	859
<u>8.53.53 imagedestroy() : détruit une image.....</u>	859
<u>8.53.54 imageellipse() : Dessine une ellipse.....</u>	859
<u>8.53.55 imagefill() : Remplit.....</u>	860
<u>8.53.56 imagefilledarc() : Dessine une ellipse partielle et la remplit.....</u>	860
<u>8.53.57 imagefilledellipse() : Dessine une ellipse pleine.....</u>	861
<u>8.53.58 imagefilledpolygon() : Dessine un polygone rempli.....</u>	861
<u>8.53.59 imagefilledrectangle() : Dessine un rectangle rempli.....</u>	862
<u>8.53.60 imagefilltoborder() : Remplit une région avec une couleur spécifique.....</u>	862
<u>8.53.61 imagefilter() : Applique un filtre à une image.....</u>	862
<u>8.53.62 imagefontheight() : Retourne la hauteur de la police.....</u>	864
<u>8.53.63 imagefontwidth() : Retourne la largeur de la police.....</u>	864
<u>8.53.64 imageftbbox() : Calcule le rectangle d'encadrement pour un texte, en utilisant la police courante et freetype2.....</u>	864
<u>8.53.65 imagefttext() : Ecrit du texte dans une image avec la police courante FreeType 2.....</u>	864
<u>8.53.66 imagegammacorrect() : Applique une correction gamma à l'image.....</u>	865
<u>8.53.67 imagegd() : Génère une image au format GD, vers le navigateur ou un fichier.....</u>	865
<u>8.53.68 imagegd2() : Génère une image au format GD2, vers le navigateur ou un fichier.....</u>	865
<u>8.53.69 imagegif() : Envoie une image GIF vers un navigateur ou un fichier.....</u>	865
<u>8.53.70 imageinterlace() : Active ou désactive l'entrelacement.....</u>	867
<u>8.53.71 imageistruecolor() : Détermine si une image est une image truecolor.....</u>	867
<u>8.53.72 imagejpeg() : Envoie une image JPEG vers un navigateur ou un fichier.....</u>	867
<u>8.53.73 imagelayereffect() : Active l'option d'alpha blending, pour utiliser les effets de libgd.....</u>	867
<u>8.53.74 imageline() : Dessine une ligne.....</u>	868
<u>8.53.75 imageloadfont() : Charge une nouvelle police.....</u>	868
<u>8.53.76 imagepalettecopy() : Copie la palette d'une image à l'autre.....</u>	869
<u>8.53.77 imagepng() : Envoie une image PNG vers un navigateur ou un fichier.....</u>	869
<u>8.53.78 imagepolygon() : Dessine un polygone.....</u>	870
<u>8.53.79 imagepsbbox() : Retourne le rectangle entourant un texte et dessiné avec</u>	

Sommaire

8 Référence des fonctions

<u>une police PostScript Type1</u>	870
<u>8.53.80 imagepscopyfont() : Fait une copie d'une police chargée pour modification ultérieure</u>	871
<u>8.53.81 imagepsencodefont() : Change le codage vectoriel d'un caractère dans une police</u>	871
<u>8.53.82 imagepsextendfont() : Etend ou condense une police de caractères</u>	872
<u>8.53.83 imagepsfreefont() : Libère la mémoire occupée par une police PostScript Type 1</u>	872
<u>8.53.84 imagepsloadfont() : Charge une police PostScript Type 1 depuis un fichier</u>	872
<u>8.53.85 imagepsslantfont() : Incline une police de caractères PostScript</u>	873
<u>8.53.86 imagepstext() : Dessine un texte sur une image avec une police PostScript Type1</u>	873
<u>8.53.87 imagerectangle() : Dessine un rectangle</u>	874
<u>8.53.88 imagerotate() : Fait tourner une image d'un angle</u>	874
<u>8.53.89 imagesavealpha() : Configure l'enregistrement des informations complètes du canal alpha lors de sauvegardes d'images PNG</u>	874
<u>8.53.90 imagesetbrush() : Modifie la brosse pour le dessin des lignes</u>	875
<u>8.53.91 imagesetpixel() : Dessine un pixel</u>	875
<u>8.53.92 imagesetstyle() : Configure le style pour le dessin des lignes</u>	875
<u>8.53.93 imagesethickness() : Modifie l'épaisseur d'un trait</u>	876
<u>8.53.94 imagesettile() : Modifie l'image utilisée pour le carrelage</u>	876
<u>8.53.95 imagestring() : Dessine une chaîne horizontale</u>	877
<u>8.53.96 imagestringup() : Dessine une chaîne verticale</u>	877
<u>8.53.97 imagesx() : Retourne la largeur d'une image</u>	877
<u>8.53.98 imagesy() : Retourne la hauteur de l'image</u>	878
<u>8.53.99 imagetruecolortopalette() : Convertit une image en vraies couleurs en image à palette</u>	878
<u>8.53.100 imagettfbbox() : Retourne le rectangle entourant un texte et dessiné avec une police TrueType</u>	878
<u>8.53.101 imagettftext() : Dessine un texte avec une police TrueType</u>	879
<u>8.53.102 imagetypes() : Retourne les types d'images supportés par la version courante de PHP</u>	881
<u>8.53.103 imagewbmp() : Affiche une image WBMP</u>	881
<u>8.53.104 imagexbm() : Génère une image au format XBM</u>	882
<u>8.53.105 iptcembed() : Intègre des données binaires IPTC dans une image JPEG</u>	882
<u>8.53.106 iptcparse() : Analyse un bloc binaire IPTC http://www.iptc.org/ et recherche les balises simples</u>	882
<u>8.53.107 jpeg2wbmp() : Convertit une image JPEG en image WBMP</u>	882
<u>8.53.108 png2wbmp() : Convertit une image PNG en image WBMP</u>	883
<u>8.54 IMAP</u>	884
<u>8.54.1 Introduction</u>	884
<u>8.54.2 Pré-requis</u>	884
<u>8.54.3 Installation</u>	884
<u>8.54.4 Configuration à l'exécution</u>	885
<u>8.54.5 Types de ressources</u>	885
<u>8.54.6 Constantes pré-définies</u>	885
<u>8.54.7 Voir aussi</u>	886
<u>8.54.9 imap_alerts() : Retourne toutes les alertes</u>	888
<u>8.54.10 imap_append() : Ajoute une chaîne dans une boîte aux lettres</u>	888
<u>8.54.11 imap_base64() : Décode un texte encodé en BASE64</u>	889
<u>8.54.12 imap_binary() : Convertit une chaîne à 8 bits en une chaîne à base64</u>	889

Sommaire

8 Référence des fonctions

<u>8.54.13 imap_body()</u> : Lit le corps d'un message.....	889
<u>8.54.14 imap_bodystruct()</u> : Lit la structure d'une section du corps d'un mail.....	890
<u>8.54.15 imap_check()</u> : Vérifie le courrier de la boîte aux lettres courante.....	890
<u>8.54.16 imap_clearflag_full()</u> : Supprime un flag (drapeau) sur un message.....	891
<u>8.54.17 imap_close()</u> : Termine un flux IMAP.....	891
<u>8.54.18 imap_createmailbox()</u> : Crée une nouvelle boîte aux lettres.....	891
<u>8.54.19 imap_delete()</u> : Marque le fichier pour l'effacement, dans la boîte aux lettres courante.....	892
<u>8.54.20 imap_deletemailbox()</u> : Efface une boîte aux lettres.....	893
<u>8.54.21 imap_errors()</u> : Retourne toutes les erreurs.....	893
<u>8.54.22 imap_expunge()</u> : Efface tous les messages marqués pour l'effacement.....	893
<u>8.54.23 imap_fetch_overview()</u> : Lit un sommaire des en-têtes de messages.....	893
<u>8.54.24 imap_fetchbody()</u> : Retourne une section extraite du corps d'un message.....	894
<u>8.54.25 imap_fetchheader()</u> : Retourne l'en-tête d'un message.....	895
<u>8.54.26 imap_fetchstructure()</u> : Lit la structure d'un message.....	895
<u>8.54.27 imap_get_quota()</u> : Lit les quotas des boîtes aux lettres.....	896
<u>8.54.28 imap_get_quotaroot()</u> : Lit les quotas de chaque utilisateur.....	897
<u>8.54.29 imap_getacl()</u> : Retourne le ACL pour la boîte aux lettres.....	898
<u>8.54.30 imap_getmailboxes()</u> : Liste les boîtes aux lettres, et retourne les détails de chacune.....	898
<u>8.54.31 imap_getsubscribed()</u> : Liste toutes les boîtes aux lettres souscrites.....	899
<u>8.54.32 imap_header()</u> : Alias de <code>imap_headerinfo</code>	899
<u>8.54.33 imap_headerinfo()</u> : Lit l'en-tête du message.....	899
<u>8.54.34 imap_headers()</u> : Retourne les en-têtes de tous les messages d'une boîte aux lettres.....	901
<u>8.54.35 imap_last_error()</u> : Retourne la dernière erreur (si elle existe) qui est survenue lors de la dernière requête.....	901
<u>8.54.36 imap_list()</u> : Lit la liste des boîtes aux lettres.....	901
<u>8.54.37 imap_listmailbox()</u> : Alias de <code>imap_list</code>	902
<u>8.54.38 imap_listscan()</u> : Lit la liste des boîtes aux lettres, et y recherche une chaîne.....	902
<u>8.54.39 imap_listsubscribed()</u> : Alias de <code>imap_lsub</code>	902
<u>8.54.40 imap_lsub()</u> : Liste toutes les boîtes aux lettres enregistrées.....	902
<u>8.54.41 imap_mail_compose()</u> : Crée un message MIME.....	902
<u>8.54.42 imap_mail_copy()</u> : Copie les messages spécifiés dans une boîte aux lettres.....	903
<u>8.54.43 imap_mail_move()</u> : Déplace des messages dans une boîte aux lettres.....	903
<u>8.54.44 imap_mail()</u> : Envoie un message mail.....	903
<u>8.54.45 imap_mailboxmsginfo()</u> : Lit les informations à propos de la boîte aux lettres courante.....	904
<u>8.54.46 imap_mime_header_decode()</u> : Décode les éléments MIME d'un en-tête.....	905
<u>8.54.47 imap_msgno()</u> : Retourne le numéro de séquence de message pour un UID donné.....	905
<u>8.54.48 imap_num_msg()</u> : Retourne le nombre de message dans la boîte aux lettres courante.....	905
<u>8.54.49 imap_num_recent()</u> : Retourne le nombre de messages récents dans la boîte aux lettres courante.....	905
<u>8.54.50 imap_open()</u> : Ouvre un flux IMAP vers une boîte aux lettres.....	906
<u>8.54.51 imap_ping()</u> : Vérifie que le flux IMAP est toujours actif.....	908
<u>8.54.52 imap_qprint()</u> : Convertit une chaîne à guillemets en une chaîne à 8 bits.....	908
<u>8.54.53 imap_renamemailbox()</u> : Renomme une boîte aux lettres.....	909

Sommaire

8 Référence des fonctions

<u>8.54.54</u>	<u>imap_reopen()</u> : Réouvre un flux IMAP vers une nouvelle boîte aux lettres	909
<u>8.54.55</u>	<u>imap_rfc822_parse_adrlist()</u> : Analyse une adresse email	909
<u>8.54.56</u>	<u>imap_rfc822_parse_headers()</u> : Analyse un en-tête mail	910
<u>8.54.57</u>	<u>imap_rfc822_write_address()</u> : Retourne une adresse email formatée correctement	910
<u>8.54.58</u>	<u>imap_scanmailbox()</u> : Alias de <u>imap_listscan</u>	910
<u>8.54.59</u>	<u>imap_search()</u> : Retourne un tableau de message après recherche	910
<u>8.54.60</u>	<u>imap_set_quota()</u> : Modifie le quota d'une boîte aux lettres	911
<u>8.54.61</u>	<u>imap_setacl()</u> : Modifie le ACL de la boîte aux lettres	912
<u>8.54.62</u>	<u>imap_setflag_full()</u> : Positionne un flag sur un message	912
<u>8.54.63</u>	<u>imap_sort()</u> : Trie des messages	912
<u>8.54.64</u>	<u>imap_status()</u> : Retourne les informations de statut sur une boîte aux lettres autres que la boîte courante	913
<u>8.54.65</u>	<u>imap_subscribe()</u> : Souscrit à une boîte aux lettres	914
<u>8.54.66</u>	<u>imap_thread()</u> : Retourne l'arbre des messages organisés par thread	914
<u>8.54.67</u>	<u>imap_timeout()</u> : Configure ou retourne le timeout	914
<u>8.54.68</u>	<u>imap_uid()</u> : Retourne l'UID d'un message	915
<u>8.54.69</u>	<u>imap_undelete()</u> : Enlève la marque d'effacement d'un message	915
<u>8.54.70</u>	<u>imap_unsubscribe()</u> : Termine la souscription à une boîte aux lettres	915
<u>8.54.71</u>	<u>imap_utf7_decode()</u> : Décode une chaîne encodée en UTF-7 modifié	915
<u>8.54.72</u>	<u>imap_utf7_encode()</u> : Convertit une chaîne ISO-8859-1 en texte UTF-7 modifié	916
<u>8.54.73</u>	<u>imap_utf8()</u> : Convertit du texte au format MIME en UTF8	916
<u>8.55</u>	<u>Options PHP et informations</u>	917
<u>8.55.1</u>	<u>Introduction</u>	917
<u>8.55.2</u>	<u>Pré-requis</u>	917
<u>8.55.3</u>	<u>Installation</u>	917
<u>8.55.4</u>	<u>Configuration à l'exécution</u>	917
<u>8.55.5</u>	<u>Types de ressources</u>	918
<u>8.55.6</u>	<u>Constantes pré-définies</u>	919
<u>8.55.8</u>	<u>assert()</u> : Vérifie si une assertion est fausse	920
<u>8.55.9</u>	<u>dl()</u> : Charge une extension PHP à la volée	921
<u>8.55.10</u>	<u>extension_loaded()</u> : Détermine si une extension est chargée ou non	923
<u>8.55.11</u>	<u>get_cfg_var()</u> : Retourne la valeur d'une option de PHP	923
<u>8.55.12</u>	<u>get_current_user()</u> : Retourne le nom du possesseur du script courant	924
<u>8.55.13</u>	<u>get_defined_constants()</u> : Retourne la liste des constantes et leurs valeurs	924
<u>8.55.14</u>	<u>get_extension_funcs()</u> : Liste les fonctions d'une extension	926
<u>8.55.15</u>	<u>get_include_path()</u> : Lit la valeur de la directive de configuration <u>include_path</u>	926
<u>8.55.16</u>	<u>get_included_files()</u> : Retourne un tableau avec les noms des fichiers qui sont inclus dans un script	926
<u>8.55.17</u>	<u>get_loaded_extensions()</u> : Retourne la liste de tous les modules compilés et chargés	927
<u>8.55.18</u>	<u>get_magic_quotes_gpc()</u> : Retourne la configuration actuelle de l'option <u>magic_quotes_gpc</u>	928
<u>8.55.19</u>	<u>get_magic_quotes_runtime()</u> : Retourne la configuration actuelle de l'option <u>magic_quotes_runtime</u>	928
<u>8.55.20</u>	<u>get_required_files()</u> : Alias de <u>get_included_files</u>	929
<u>8.55.21</u>	<u>getenv()</u> : Retourne la valeur d'une variable d'environnement	929
<u>8.55.22</u>	<u>getlastmod()</u> : Retourne la date de dernière modification de la page	929

Sommaire

8 Référence des fonctions

<u>8.55.23 getmygid()</u> : Retourne le GID du propriétaire du script.....	930
<u>8.55.24 getmyinode()</u> : Retourne l'inode du script.....	930
<u>8.55.25 getmypid()</u> : Retourne le numéro de processus courant de PHP.....	930
<u>8.55.26 getmyuid()</u> : Retourne l'UID du propriétaire du script actuel.....	930
<u>8.55.27 getopt()</u> : Lit des options passées dans la ligne de commande.....	930
<u>8.55.28 getrusage()</u> : Retourne le niveau d'utilisation des ressources.....	931
<u>8.55.29 ini_alter()</u> : Alias de ini_set.....	931
<u>8.55.30 ini_get_all()</u> : Lit toutes les valeurs de configuration.....	931
<u>8.55.31 ini_get()</u> : Lit la valeur d'une option de configuration.....	932
<u>8.55.32 ini_restore()</u> : Restaure la valeur de l'option de configuration.....	933
<u>8.55.33 ini_set()</u> : Modifie la valeur d'une option de configuration.....	934
<u>8.55.34 main()</u> : Fausse documentation pour main.....	934
<u>8.55.35 memory_get_usage()</u> : Indique la quantité de mémoire utilisée par PHP.....	934
<u>8.55.36 php_ini_scanned_files()</u> : Retourne la liste des fichiers .ini analysés dans les dossiers de configuration supplémentaires.....	935
<u>8.55.37 php_logo_guid()</u> : Retourne l'identifiant du logo PHP.....	935
<u>8.55.38 php_sapi_name()</u> : Retourne le type d'interface utilisée entre le serveur web et PHP.....	936
<u>8.55.39 php_uname()</u> : Retourne les informations sur le système d'exploitation.....	936
<u>8.55.40 phpcredits()</u> : Affiche les crédits de PHP.....	937
<u>8.55.41 phpinfo()</u> : Affiche de nombreuses informations sur PHP.....	938
<u>8.55.42 phpversion()</u> : Retourne le numéro de la version courante de PHP.....	939
<u>8.55.43 putenv()</u> : Fixe la valeur d'une variable d'environnement.....	940
<u>8.55.44 restore_include_path()</u> : Restaure la valeur de la directive de configuration include_path.....	940
<u>8.55.45 set_include_path()</u> : Modifie la valeur de la directive de configuration include_path.....	941
<u>8.55.46 set_magic_quotes_runtime()</u> : Active/désactive l'option magic_quotes_runtime.....	941
<u>8.55.47 set_time_limit()</u> : Fixe le temps maximum d'exécution d'un script.....	941
<u>8.55.48 version_compare()</u> : Compare deux chaînes de versions au format des versions PHP.....	942
<u>8.55.49 zend_logo_guid()</u> : Retourne le logo de Zend.....	943
<u>8.55.50 zend_version()</u> : Lit la version courante du moteur Zend.....	943
<u>8.56 Ingres II</u>	944
<u>8.56.1 Introduction</u>	944
<u>8.56.2 Pré-requis</u>	944
<u>8.56.3 Installation</u>	944
<u>8.56.4 Configuration à l'exécution</u>	944
<u>8.56.5 Types de ressources</u>	945
<u>8.56.6 Constantes pré-définies</u>	945
<u>8.56.8 ingres_close()</u> : Ferme une connexion à un serveur Ingres.....	947
<u>8.56.9 ingres_commit()</u> : Valide une transaction.....	947
<u>8.56.10 ingres_connect()</u> : Ouvre une connexion à un serveur Ingres.....	947
<u>8.56.11 ingres_cursor()</u> : Récupère le nom du curseur pour une ressource donnée.....	948
<u>8.56.12 ingres_errno()</u> : Récupère le dernier numéro d'erreur d'ingres généré.....	948
<u>8.56.13 ingres_error()</u> : Récupère un message d'erreur significatif pour la dernière erreur générée.....	949
<u>8.56.14 ingres_errsqlstate()</u> : Gets the last SQLSTATE error code generated.....	949
<u>8.56.15 ingres_fetch_array()</u> : Récupère une ligne de résultat Ingres dans un tableau.....	950

Sommaire

8 Référence des fonctions

<u>8.56.16</u> <u>ingres_fetch_object()</u> : Récupère une ligne de résultat dans un objet.....	951
<u>8.56.17</u> <u>ingres_fetch_row()</u> : Récupère une ligne de résultat dans un tableau énuméré.....	951
<u>8.56.18</u> <u>ingres_field_length()</u> : Retourne la taille d'un champ.....	952
<u>8.56.19</u> <u>ingres_field_name()</u> : Retourne le nom d'un champ dans le résultat d'une requête.....	952
<u>8.56.20</u> <u>ingres_field_nullable()</u> : Teste si un champ est annulable.....	952
<u>8.56.21</u> <u>ingres_field_precision()</u> : Retourne la précision d'un champ.....	953
<u>8.56.22</u> <u>ingres_field_scale()</u> : Retourne l'échelle d'un champ.....	953
<u>8.56.23</u> <u>ingres_field_type()</u> : Retourne le type d'un champ dans le résultat d'une requête.....	953
<u>8.56.24</u> <u>ingres_num_fields()</u> : Retourne le nombre de champs renvoyés par la dernière requête.....	954
<u>8.56.25</u> <u>ingres_num_rows()</u> : Retourne le nombre de lignes affectées ou renvoyées par la dernière requête.....	954
<u>8.56.26</u> <u>ingres_pconnect()</u> : Ouvre une connexion persistante à un serveur Ingres.....	954
<u>8.56.27</u> <u>ingres_query()</u> : Envoie une requête SQL à un serveur Ingres II.....	955
<u>8.56.28</u> <u>ingres_rollback()</u> : Annule une transaction.....	955
<u>8.57</u> <u>IRC</u>	956
<u>8.57.1</u> <u>Introduction</u>	956
<u>8.57.2</u> <u>Installation</u>	956
<u>8.57.3</u> <u>Configuration à l'exécution</u>	956
<u>8.57.4</u> <u>Types de ressources</u>	956
<u>8.57.5</u> <u>Constantes pré-définies</u>	956
<u>8.57.7</u> <u>ircg_disconnect()</u> : Ferme la connexion avec un serveur.....	957
<u>8.57.8</u> <u>ircg_eval_ecmascript_params()</u> : Décode une liste de paramètres JS-encodés.....	957
<u>8.57.9</u> <u>ircg_fetch_error_msg()</u> : Retourne l'erreur rencontrée par la dernière opération IRCG.....	958
<u>8.57.10</u> <u>ircg_get_username()</u> : Retourne le nom d'utilisateur pour la connexion.....	958
<u>8.57.11</u> <u>ircg_html_encode()</u> : Prépare des données IRC pour l'affichage HTML.....	958
<u>8.57.12</u> <u>ircg_ignore_add()</u> : Ajoute un utilisateur sur la liste des utilisateurs indésirables.....	959
<u>8.57.13</u> <u>ircg_ignore_del()</u> : Supprime un utilisateur de la liste des utilisateurs indésirables.....	959
<u>8.57.14</u> <u>ircg_invite()</u> : Invite un utilisateur à rejoindre un canal.....	959
<u>8.57.15</u> <u>ircg_is_conn_alive()</u> : Vérifie l'état de la connexion.....	959
<u>8.57.16</u> <u>ircg_join()</u> : Rejoint un canal IRC.....	959
<u>8.57.17</u> <u>ircg_kick()</u> : Expulse un utilisateur d'un canal.....	959
<u>8.57.18</u> <u>ircg_list()</u> : Liste le sujet et le nombre d'utilisateurs d'un ou plusieurs canaux.....	960
<u>8.57.19</u> <u>ircg_lookup_format_messages()</u> : Sélectionne un format d'affichage pour les messages IRC.....	960
<u>8.57.20</u> <u>ircg_lusers()</u> : Statistiques réseau IRC.....	961
<u>8.57.21</u> <u>ircg_msg()</u> : Envoie un message à un canal ou un utilisateur.....	961
<u>8.57.22</u> <u>ircg_names()</u> : Récupère les noms d'utilisateurs visibles.....	961
<u>8.57.23</u> <u>ircg_nick()</u> : Change de nom sur le serveur.....	961
<u>8.57.24</u> <u>ircg_nickname_escape()</u> : Encode les caractères spéciaux dans un pseudo pour être compatible avec IRC.....	961
<u>8.57.25</u> <u>ircg_nickname_unescape()</u> : Décode un pseudo encodé.....	962
<u>8.57.26</u> <u>ircg_notice()</u> : Envoie une note (notice) à un utilisateur.....	962

Sommaire

8 Référence des fonctions

<u>8.57.27 ircg_oper()</u> : Elève les privilèges au rang d'opérateur IRC.....	962
<u>8.57.28 ircg_part()</u> : Quitte le canal.....	962
<u>8.57.29 ircg_pconnect()</u> : Connecte à un serveur IRC.....	962
<u>8.57.30 ircg_register_format_messages()</u> : Enregistre un nouveau format d'affichage des messages IRC.....	963
<u>8.57.31 ircg_set_current()</u> : Prépare la connexion courante pour l'affichage.....	964
<u>8.57.32 ircg_set_file()</u> : Configure le journal de connexion.....	964
<u>8.57.33 ircg_set_on_die()</u> : Configure une action à exécuter lorsque la connexion IRC se termine.....	964
<u>8.57.34 ircg_topic()</u> : Modifie le sujet (topic) d'un canal.....	964
<u>8.57.35 ircg_who()</u> : Interroge le serveur sur les informations WHO.....	965
<u>8.57.36 ircg_whois()</u> : Requier les informations sur un utilisateur.....	965
<u>8.58 Intégration de Java avec PHP</u>	966
<u>8.58.1 Introduction</u>	966
<u>8.58.2 Pré-requis</u>	966
<u>8.58.3 Installation</u>	966
<u>8.58.4 Configuration à l'exécution</u>	966
<u>8.58.5 Types de ressources</u>	967
<u>8.58.6 Constantes pré-définies</u>	967
<u>8.58.7 Exemples</u>	967
<u>8.58.8 Java Servlet SAPI</u>	968
<u>8.58.10 java_last_exception_get()</u> : Lit la dernière exception Java.....	969
<u>8.59 KADM5</u>	970
<u>8.59.1 Introduction</u>	970
<u>8.59.2 Constantes pré-définies</u>	970
<u>8.59.3 Exemples</u>	971
<u>8.59.4 Contacts</u>	972
<u>8.59.6 kadm5_create_principal()</u> : Crée un principal kerberos avec les paramètres donnés.....	972
<u>8.59.7 kadm5_delete_principal()</u> : Efface un principal kerberos.....	972
<u>8.59.8 kadm5_destroy()</u> : Ferme la connexion avec le serveur d'administration et libère toutes les ressources associées.....	973
<u>8.59.9 kadm5_flush()</u> : Valide toutes les modifications de la base de données Kerberos, et quitte la connexion ouverte du serveur d'administration Kerberos.....	973
<u>8.59.10 kadm5_get_policies()</u> : Récupère toutes les directives depuis la base de données Kerberos.....	973
<u>8.59.11 kadm5_get_principal()</u> : Récupère les entrées des éléments principaux depuis la base de données Kerberos.....	973
<u>8.59.12 kadm5_get_principals()</u> : Récupère tous les éléments principaux depuis la base de données Kerberos.....	973
<u>8.59.13 kadm5_init_with_password()</u> : Ouvre une connexion à la bibliothèque KADM5 et initialise toutes les informations nécessaires.....	973
<u>8.59.14 kadm5_modify_principal()</u> : Modifie un principal Kerberos avec les paramètres donnés.....	974
<u>8.60 LDAP</u>	975
<u>8.60.1 Introduction</u>	975
<u>8.60.2 Pré-requis</u>	976
<u>8.60.3 Installation</u>	976
<u>8.60.4 Configuration à l'exécution</u>	976
<u>8.60.5 Types de ressources</u>	976
<u>8.60.6 Constantes pré-définies</u>	976

Sommaire

8 Référence des fonctions

<u>8.60.7 Exemples</u>	977
<u>8.60.8 Utiliser les fonctions LDAP de PHP</u>	978
<u>8.60.10 ldap_add() : Ajoute une entrée dans un dossier LDAP</u>	979
<u>8.60.11 ldap_bind() : Authentification au serveur LDAP</u>	980
<u>8.60.12 ldap_close() : Alias de ldap_unbind</u>	981
<u>8.60.13 ldap_compare() : Compare une entrée avec des valeurs d'attributs</u>	981
<u>8.60.14 ldap_connect() : Se connecte à un serveur LDAP</u>	982
<u>8.60.15 ldap_count_entries() : Compte le nombre d'entrées après une recherche</u>	983
<u>8.60.16 ldap_delete() : Efface une entrée dans un dossier</u>	983
<u>8.60.17 ldap_dn2ufn() : Convertit un DN en format UFN (User Friendly Naming)</u>	983
<u>8.60.18 ldap_err2str() : Convertit un numéro d'erreur LDAP en message d'erreur</u>	983
<u>8.60.19 ldap_errno() : Retourne le numéro d'erreur LDAP de la dernière commande exécutée</u>	984
<u>8.60.20 ldap_error() : Retourne le message LDAP de la dernière commande LDAP</u>	984
<u>8.60.21 ldap_explode_dn() : Sépare les différents composants d'un DN</u>	985
<u>8.60.22 ldap_first_attribute() : Retourne le premier attribut</u>	985
<u>8.60.23 ldap_first_entry() : Retourne la première entrée</u>	985
<u>8.60.24 ldap_first_reference() : Retourne la première référence</u>	985
<u>8.60.25 ldap_free_result() : Libère la mémoire du résultat</u>	986
<u>8.60.26 ldap_get_attributes() : Lit les attributs d'une entrée</u>	986
<u>8.60.27 ldap_get_dn() : Lit le DN d'une entrée</u>	987
<u>8.60.28 ldap_get_entries() : Lit toutes les entrées du résultat</u>	987
<u>8.60.29 ldap_get_option() : Lit/écrit la valeur courante d'une option</u>	987
<u>8.60.30 ldap_get_values_len() : Lit toutes les valeurs binaires d'une entrée</u>	988
<u>8.60.31 ldap_get_values() : Lit toutes les valeurs d'une entrée LDAP</u>	988
<u>8.60.32 ldap_list() : Recherche dans un niveau</u>	989
<u>8.60.33 ldap_mod_add() : Ajoute un attribut à l'entrée courante</u>	990
<u>8.60.34 ldap_mod_del() : Efface un attribut à l'entrée courante</u>	990
<u>8.60.35 ldap_mod_replace() : Remplace un attribut dans l'entrée courante</u>	990
<u>8.60.36 ldap_modify() : Modifie une entrée LDAP</u>	990
<u>8.60.37 ldap_next_attribute() : Lit l'attribut suivant</u>	991
<u>8.60.38 ldap_next_entry() : Lit la prochaine entrée</u>	991
<u>8.60.39 ldap_next_reference() : Lit la référence suivante</u>	991
<u>8.60.40 ldap_parse_reference() : Extrait les informations d'une référence d'entrée</u>	991
<u>8.60.41 ldap_parse_result() : Extrait des informations d'un résultat</u>	991
<u>8.60.42 ldap_read() : Lit une entrée</u>	992
<u>8.60.43 ldap_rename() : Modifie le nom d'une entrée</u>	992
<u>8.60.44 ldap_sasl_bind() : Authentification au serveur LDAP en utilisant SASL</u>	992
<u>8.60.45 ldap_search() : Recherche sur le serveur LDAP</u>	993
<u>8.60.46 ldap_set_option() : Modifie la valeur d'une option LDAP</u>	994
<u>8.60.47 ldap_set_rebind_proc() : Configure une fonction de callback pour refaire des liaisons lors de recherche de référants</u>	995
<u>8.60.48 ldap_sort() : Trie les entrées d'un résultat LDAP</u>	995
<u>8.60.49 ldap_start_tls() : Démarre TLS</u>	995
<u>8.60.50 ldap_t61_to_8859() : Convertit les caractères t6 en caractères 8859</u>	996
<u>8.60.51 ldap_unbind() : Déconnecte d'un serveur LDAP</u>	996
<u>8.61 libxml</u>	997
<u>8.61.1 Introduction</u>	997
<u>8.61.2 Pré-requis</u>	997
<u>8.61.3 Classes pré-définies</u>	997

Sommaire

8 Référence des fonctions

<u>8.61.4 Constantes pré-définies</u>	997
<u>8.61.6 libxml_get_errors()</u> : Lit le tableau d'erreurs.....	998
<u>8.61.7 libxml_get_last_error()</u> : Lit la dernière erreur libxml.....	999
<u>8.61.8 libxml_set_streams_context()</u> : Configure le contexte de flux pour la <u>prochaine opération libxml</u>	999
<u>8.61.9 libxml_use_internal_errors()</u> : Désactive le rapport d'erreur libxml et les <u>stocke pour lecture ultérieure</u>	999
<u>8.62 Fonctions LZF</u>	1000
<u>8.62.1 Introduction</u>	1000
<u>8.62.2 Installation</u>	1000
<u>8.62.4 lzf_decompress()</u> : Décompression LZF.....	1000
<u>8.62.5 lzf_optimized_for()</u> : Détermine le mode d'optimisation de l'extension LZF.....	1000
<u>8.63 Mail</u>	1001
<u>8.63.1 Introduction</u>	1001
<u>8.63.2 Pré-requis</u>	1001
<u>8.63.3 Installation</u>	1001
<u>8.63.4 Configuration à l'exécution</u>	1001
<u>8.63.5 Types de ressources</u>	1002
<u>8.63.6 Constantes pré-définies</u>	1002
<u>8.63.8 mail()</u> : Envoi un mail.....	1002
<u>8.64 Traitement d'email</u>	1003
<u>8.64.1 Introduction</u>	1003
<u>8.64.2 Installation</u>	1003
<u>8.64.4 mailparse_msg_create()</u> : Retourne un analyseur destiné au traitement <u>d'un message</u>	1003
<u>8.64.5 mailparse_msg_extract_part_file()</u> : Extrait et décode une section de <u>message</u>	1004
<u>8.64.6 mailparse_msg_extract_part()</u> : Extrait et décode une section de message.....	1004
<u>8.64.7 mailparse_msg_free()</u> : Détruit un analyseur créé par <u>mailparse_msg_create</u>	1004
<u>8.64.8 mailparse_msg_get_part_data()</u> : Retourne un tableau associatif avec des <u>informations sur le message</u>	1005
<u>8.64.9 mailparse_msg_get_part()</u> : Retourne une ressource sur une section d'un <u>message MIME</u>	1005
<u>8.64.10 mailparse_msg_get_structure()</u> : Retourne un tableau avec les <u>différentes sections MIME du message</u>	1005
<u>8.64.11 mailparse_msg_parse_file()</u> : Analyse un fichier et retourne une <u>ressource représentant la structure du fichier</u>	1005
<u>8.64.12 mailparse_msg_parse()</u> : Analyse incrémentalement des données dans <u>un buffer</u>	1006
<u>8.64.13 mailparse_rfc822_parse_addresses()</u> : Analyse les adresses et retourne <u>un tableau contenant ces données</u>	1006
<u>8.64.14 mailparse_stream_encode()</u> : Lit les données dans un fichier, applique <u>l'encodage et envoie le résultat à destfp</u>	1006
<u>8.64.15 mailparse_uudecode_all()</u> : Scanne les données du fichier et extrait tous <u>les fichiers encodés qui s'y trouvent</u>	1006
<u>8.65 Mathématiques</u>	1008
<u>8.65.1 Introduction</u>	1008
<u>8.65.2 Pré-requis</u>	1008
<u>8.65.3 Installation</u>	1008
<u>8.65.4 Configuration à l'exécution</u>	1008

Sommaire

8 Référence des fonctions

<u>8.65.5 Types de ressources</u>	1008
<u>8.65.6 Constantes pré-définies</u>	1008
<u>8.65.8 acos() : arc cosinus</u>	1010
<u>8.65.9 acosh() : Arc cosinus hyperbolique</u>	1010
<u>8.65.10 asin() : arc sinus</u>	1010
<u>8.65.11 asinh() : Arc sinus hyperbolique</u>	1010
<u>8.65.12 atan() : arc tangent</u>	1011
<u>8.65.13 atan2() : Arc tangent de deux variables</u>	1011
<u>8.65.14 atanh() : Arc tangeant hyperbolique</u>	1011
<u>8.65.15 base_convert() : Convertit un nombre entre des bases arbitraires</u>	1011
<u>8.65.16 bindec() : Convertit de binaire en décimal</u>	1012
<u>8.65.17 ceil() : Arrondit au nombre supérieur</u>	1012
<u>8.65.18 cos() : cosinus</u>	1013
<u>8.65.19 cosh() : Cosinus hyperbolic</u>	1013
<u>8.65.20 decbin() : Convertit de décimal en binaire</u>	1013
<u>8.65.21 dechex() : Convertit de décimal en hexadécimal</u>	1013
<u>8.65.22 decoct() : Convertit de décimal en octal</u>	1014
<u>8.65.23 deg2rad() : Convertit un nombre de degrés en radians</u>	1014
<u>8.65.24 exp() : Exponentielle</u>	1015
<u>8.65.25 expm1() : Retourne le résultat de exp(number) - 1, calculé de manière précise, même si val est proche de 0</u>	1015
<u>8.65.26 floor() : Arrondit à l'entier inférieur</u>	1015
<u>8.65.27 fmod() : Retourne le reste de la division</u>	1016
<u>8.65.28 getrandmax() : Plus grande valeur aléatoire possible</u>	1016
<u>8.65.29 hexdec() : Convertit de hexadécimal en décimal</u>	1016
<u>8.65.30 hypot() : Calcul la longueur de l'hypoténuse d'un triangle à angle droit</u>	1017
<u>8.65.31 is_finite() : Indique si un nombre est fini</u>	1017
<u>8.65.32 is_infinite() :</u>	1017
<u>8.65.33 is_nan() : Indique si une valeur est un nombre</u>	1017
<u>8.65.34 lcg_value() : Générateur de congruence combinée linéaire</u>	1017
<u>8.65.35 log() : Logarithme naturel (népérien)</u>	1017
<u>8.65.36 log10() : logarithme en base 10</u>	1018
<u>8.65.37 log1p() : Retourne le résultat de log(1 + number), calculé de manière précise, même si val est proche de zéro</u>	1018
<u>8.65.38 max() : La plus grande valeur</u>	1018
<u>8.65.39 min() : La plus petite valeur</u>	1019
<u>8.65.40 mt_getrandmax() : La plus grand valeur aléatoire possible</u>	1020
<u>8.65.41 mt_rand() : Génère une valeur aléatoire (meilleure méthode)</u>	1020
<u>8.65.42 mt_srand() : Initialise une valeur aléatoire (meilleure méthode)</u>	1020
<u>8.65.43 octdec() : Conversion d'octal en décimal</u>	1021
<u>8.65.44 pi() : Retourne la valeur de pi</u>	1021
<u>8.65.45 pow() : Expression exponentielle</u>	1022
<u>8.65.46 rad2deg() : Conversion de radians en degrés</u>	1022
<u>8.65.47 rand() : Génère une valeur aléatoire</u>	1022
<u>8.65.48 round() : Arrondi</u>	1023
<u>8.65.49 sin() : Sinus</u>	1024
<u>8.65.50 sinh() : Sinyus hyperbolique</u>	1024
<u>8.65.51 sqrt() : Racine carrée</u>	1024
<u>8.65.52 srand() : Initialise le générateur de nombres aléatoires</u>	1024
<u>8.65.53 tan() : Tangente</u>	1025
<u>8.65.54 tanh() : Tangente hyperbolique</u>	1025

Sommaire

8 Référence des fonctions

<u>8.66 MaxDB PHP Extension</u>	1026
<u>8.66.1 Introduction</u>	1026
<u>8.66.2 Pré-requis</u>	1026
<u>8.66.3 Installation</u>	1027
<u>8.66.4 Configuration à l'exécution</u>	1027
<u>8.66.5 Classes pré-définies</u>	1027
<u>8.66.6 Constantes pré-définies</u>	1030
<u>8.66.7 Exemples</u>	1031
<u>8.66.9 maxdb autocommit() : maxdb->auto_commit()</u>	1033
<u>8.66.10 maxdb bind_param() : Alias de maxdb_stmt_bind_param</u>	1034
<u>8.66.11 maxdb bind_result() : Alias de maxdb_stmt_bind_result</u>	1034
<u>8.66.12 maxdb change_user() : maxdb->change_user()</u>	1034
<u>8.66.13 maxdb character set name() : maxdb->character set name()</u>	1034
<u>8.66.14 maxdb client_encoding() : Alias de maxdb_character set name</u>	1035
<u>8.66.15 maxdb close_long_data() : maxdb->close_long_data()</u>	1035
<u>8.66.16 maxdb close() : maxdb->close()</u>	1035
<u>8.66.17 maxdb commit() : maxdb->commit()</u>	1035
<u>8.66.18 maxdb connect_errno() : Retourne le code erreur du dernier appel à la connexion</u>	1035
<u>8.66.19 maxdb connect_error() : Retourne une chaîne descriptive de la dernière erreur</u>	1036
<u>8.66.20 maxdb connect() : maxdb()</u>	1036
<u>8.66.21 maxdb data_seek() : result->data_seek()</u>	1036
<u>8.66.22 maxdb debug() : Effectue des opérations de débogage</u>	1037
<u>8.66.23 maxdb_disable_reads_from_master() : maxdb->disable_reads_from_master()</u>	1037
<u>8.66.24 maxdb disable_rpl_parse() : Désactive l'analyseur RPL</u>	1037
<u>8.66.25 maxdb dump_debug_info() : Sauvegarde les informations de débogage dans un log</u>	1037
<u>8.66.26 maxdb embedded_connect() : Ouvre une connexion sur un serveur MaxDB embarqué</u>	1037
<u>8.66.27 maxdb enable_reads_from_master() : Active la lecture depuis le maître</u>	1038
<u>8.66.28 maxdb enable_rpl_parse() : Active l'analyseur RPL</u>	1038
<u>8.66.29 maxdb errno() : maxdb->errno()</u>	1038
<u>8.66.30 maxdb error() : Retourne une chaîne représentant la dernière erreur</u>	1038
<u>8.66.31 maxdb escape_string() : Alias de maxdb_real_escape_string</u>	1039
<u>8.66.32 maxdb execute() : Alias de maxdb_stmt_execute</u>	1039
<u>8.66.33 maxdb fetch_array() : result->fetch_array()</u>	1039
<u>8.66.34 maxdb fetch_assoc() : maxdb->fetch_assoc()</u>	1040
<u>8.66.35 maxdb fetch_field_direct() : result->fetch_field_direct()</u>	1040
<u>8.66.36 maxdb fetch_field() : result->fetch_field()</u>	1040
<u>8.66.37 maxdb fetch_fields() : result->fetch_fields()</u>	1041
<u>8.66.38 maxdb fetch_lengths() : result->lengths()</u>	1041
<u>8.66.39 maxdb fetch_object() : result->fetch_object()</u>	1041
<u>8.66.40 maxdb fetch_row() : result->fetch_row()</u>	1042
<u>8.66.41 maxdb fetch() : Alias de maxdb_stmt_fetch</u>	1042
<u>8.66.42 maxdb field_count() : maxdb->field_count()</u>	1042
<u>8.66.43 maxdb field_seek() : result->field_seek()</u>	1042
<u>8.66.44 maxdb field_tell() : result->current_field()</u>	1043
<u>8.66.45 maxdb free_result() : result->free()</u>	1043
<u>8.66.46 maxdb get_client_info() : Retourne la version du client MaxDB sous la</u>	

Sommaire

8 Référence des fonctions

<u>forme d'une chaîne</u>	1043
<u>8.66.47 maxdb_get_client_version() : Récupère des informations sur le client MaxDB</u>	1043
<u>8.66.48 maxdb_get_host_info() : maxdb->get_host_info()</u>	1044
<u>8.66.49 maxdb_get_metadata() : Alias de maxdb_stmt_result_metadata</u>	1044
<u>8.66.50 maxdb_get_proto_info() : maxdb->protocol_version()</u>	1044
<u>8.66.51 maxdb_get_server_info() : maxdb->server_info()</u>	1044
<u>8.66.52 maxdb_get_server_version() : Retourne la version du serveur MaxDB sous la forme d'un entier</u>	1045
<u>8.66.53 maxdb_info() : maxdb->info()</u>	1045
<u>8.66.54 maxdb_init() : Initialise MaxDB et retourne une ressource pour l'utiliser avec maxdb_real_connect</u>	1045
<u>8.66.55 maxdb_insert_id() : maxdb->insert_id()</u>	1046
<u>8.66.56 maxdb_kill() : maxdb->kill()</u>	1046
<u>8.66.57 maxdb_master_query() : Force l'exécution d'une requête sur le maître dans une configuration maître/esclave</u>	1046
<u>8.66.58 maxdb_more_results() : maxdb->more_results()</u>	1046
<u>8.66.59 maxdb_multi_query() : maxdb->multi_query()</u>	1047
<u>8.66.60 maxdb_next_result() : maxdb->next_result()</u>	1047
<u>8.66.61 maxdb_num_fields() : result->field_count()</u>	1047
<u>8.66.62 maxdb_num_rows() : Récupère le nombre de lignes dans un résultat</u>	1047
<u>8.66.63 maxdb_options() : maxdb->options()</u>	1048
<u>8.66.64 maxdb_param_count() : Alias de maxdb_stmt_param_count</u>	1048
<u>8.66.65 maxdb_ping() : maxdb->ping()</u>	1049
<u>8.66.66 maxdb_prepare() : maxdb->prepare()</u>	1049
<u>8.66.67 maxdb_query() : maxdb->query()</u>	1050
<u>8.66.68 maxdb_real_connect() : maxdb->real_connect()</u>	1050
<u>8.66.69 maxdb_real_escape_string() : maxdb->real_escape_string()</u>	1050
<u>8.66.70 maxdb_real_query() : maxdb->real_query()</u>	1051
<u>8.66.71 maxdb_report() : Active ou désactive le rapport interne des fonctions</u>	1051
<u>8.66.72 maxdb_rollback() : maxdb->rollback()</u>	1051
<u>8.66.73 maxdb_rpl_parse_enabled() : Vérifie si l'analyseur RPL est activé</u>	1051
<u>8.66.74 maxdb_rpl_probe() : Test RPL</u>	1051
<u>8.66.75 maxdb_rpl_query_type() : maxdb->rpl_query_type()</u>	1052
<u>8.66.76 maxdb_select_db() : maxdb->select_db()</u>	1052
<u>8.66.77 maxdb_send_long_data() : Alias de maxdb_stmt_send_long_data</u>	1052
<u>8.66.78 maxdb_send_query() : maxdb->send_query()</u>	1052
<u>8.66.79 maxdb_server_end() : Eteint un serveur embarqué</u>	1052
<u>8.66.80 maxdb_server_init() : Initialise un serveur embarqué</u>	1053
<u>8.66.81 maxdb_set_opt() : Alias de maxdb_options</u>	1053
<u>8.66.82 maxdb_sqlstate() : maxdb->sqlstate()</u>	1053
<u>8.66.83 maxdb_ssl_set() : maxdb->ssl_set()</u>	1053
<u>8.66.84 maxdb_stat() : maxdb->stat()</u>	1053
<u>8.66.85 maxdb_stmt_affected_rows() : maxdb_stmt->affected_rows()</u>	1054
<u>8.66.86 maxdb_stmt_bind_param() : stmt->bind_param()</u>	1054
<u>8.66.87 maxdb_stmt_bind_result() : stmt->bind_result()</u>	1055
<u>8.66.88 maxdb_stmt_close_long_data() : stmt->close_long_data()</u>	1055
<u>8.66.89 maxdb_stmt_close() : maxdb_stmt->close()</u>	1056
<u>8.66.90 maxdb_stmt_data_seek() : stmt->data_seek()</u>	1056
<u>8.66.91 maxdb_stmt_errno() : maxdb_stmt->errno()</u>	1056
<u>8.66.92 maxdb_stmt_error() : maxdb_stmt->error()</u>	1057

Sommaire

8 Référence des fonctions

<u>8.66.93 maxdb_stmt_execute() : stmt->execute()</u>	1057
<u>8.66.94 maxdb_stmt_fetch() : stmt->fetch()</u>	1057
<u>8.66.95 maxdb_stmt_free_result() : stmt->free_result()</u>	1058
<u>8.66.96 maxdb_stmt_init() : maxdb->stmt_init()</u>	1058
<u>8.66.97 maxdb_stmt_num_rows() : stmt->num_rows()</u>	1058
<u>8.66.98 maxdb_stmt_param_count() : stmt->param_count()</u>	1058
<u>8.66.99 maxdb_stmt_prepare() : stmt->prepare()</u>	1059
<u>8.66.100 maxdb_stmt_reset() : stmt->reset()</u>	1059
<u>8.66.101 maxdb_stmt_result_metadata() : Retourne les méta-données du jeu de résultats pour une requête préparée</u>	1060
<u>8.66.102 maxdb_stmt_send_long_data() : stmt->send_long_data()</u>	1060
<u>8.66.103 maxdb_stmt_sqlstate() : Retourne une erreur SQLSTATE depuis la requête précédente</u>	1061
<u>8.66.104 maxdb_stmt_store_result() : maxdb->store_result()</u>	1061
<u>8.66.105 maxdb_store_result() : maxdb->store_result()</u>	1061
<u>8.66.106 maxdb_thread_id() : maxdb->thread_id()</u>	1061
<u>8.66.107 maxdb_thread_safe() : Indique si le thread safety est donné ou non</u>	1062
<u>8.66.108 maxdb_use_result() : maxdb->use_result()</u>	1062
<u>8.66.109 maxdb_warning_count() : maxdb->warning_count()</u>	1062
<u>8.67 Chaînes de caractères multi-octets</u>	1063
<u>8.67.1 Introduction</u>	1063
<u>8.67.2 Installation</u>	1064
<u>8.67.3 Configuration à l'exécution</u>	1064
<u>8.67.4 Types de ressources</u>	1067
<u>8.67.5 Constantes pré-définies</u>	1067
<u>8.67.6 Entrées/Sorties HTTP</u>	1067
<u>8.67.7 Jeux de caractères supportés</u>	1068
<u>8.67.8 Exploitation des chaînes multi-octets en PHP</u>	1069
<u>8.67.9 Cas des caractères japonais</u>	1070
<u>8.67.10 Références</u>	1071
<u>8.67.11 Jeux de caractères supportés</u>	1071
<u>8.67.13 mb_convert_encoding() : Conversion d'encodage</u>	1073
<u>8.67.14 mb_convert_kana() : Convertit entre les différents "kana"</u>	1073
<u>8.67.15 mb_convert_variables() : Convertit l'encodage de variables</u>	1074
<u>8.67.16 mb_decode_mimeheader() : Décode un en-tête MIME</u>	1075
<u>8.67.17 mb_decode_numericentity() : Décode les entités HTML en caractères</u>	1075
<u>8.67.18 mb_detect_encoding() : Détecte un encodage</u>	1075
<u>8.67.19 mb_detect_order() : Lit/modifie l'ordre de détection des encodages</u>	1076
<u>8.67.20 mb_encode_mimeheader() : Encode une chaîne pour un en-tête MIME</u>	1077
<u>8.67.21 mb_encode_numericentity() : Encode des entités HTML</u>	1078
<u>8.67.22 mb_ereg_match() : Expression rationnelle POSIX pour les chaînes multi-octets</u>	1078
<u>8.67.23 mb_ereg_replace() : Remplace des segments de chaînes, avec le support des expressions rationnelles multi-octets</u>	1079
<u>8.67.24 mb_ereg_search_getpos() : Retourne l'offset du début du prochain segment repéré par une expression rationnelle</u>	1079
<u>8.67.25 mb_ereg_search_getregs() : Lit le dernier segment de chaîne multi-octets qui correspond au masque</u>	1079
<u>8.67.26 mb_ereg_search_init() : Configure les chaînes et les expressions rationnelles pour le support des caractères multi-octets</u>	1080
<u>8.67.27 mb_ereg_search_pos() : Retourne la position et la longueur du segment</u>	

Sommaire

8 Référence des fonctions

<u>de chaîne qui vérifie le masque de l'expression rationnelle</u>	1080
<u>8.67.28 mb_ereg_search_regs() : Retourne le segment de chaîne trouvé par une expression rationnelle multi-octets</u>	1080
<u>8.67.29 mb_ereg_search_setpos() : Choisit le point de départ de la recherche par expression rationnelle</u>	1080
<u>8.67.30 mb_ereg_search() : Recherche par expression rationnelle multi-octets</u>	1081
<u>8.67.31 mb_ereg() : Recherche par expression rationnelle avec support des caractères multi-octets</u>	1081
<u>8.67.32 mb_eregi_replace() : Expression rationnelle avec support des caractères multi-octets, sans tenir compte de la casse</u>	1081
<u>8.67.33 mb_eregi() : Expression rationnelle insensible à la casse avec le support des caractères multi-octets</u>	1082
<u>8.67.34 mb_get_info() : Lit la configuration interne de l'extension mbstring</u>	1082
<u>8.67.35 mb_http_input() : Détecte le type d'encodage d'un caractère HTTP</u>	1082
<u>8.67.36 mb_http_output() : Lit/modifie l'encodage d'affichage</u>	1083
<u>8.67.37 mb_internal_encoding() : Lit/modifie l'encodage interne</u>	1083
<u>8.67.38 mb_language() : Lit/modifie le langage courant</u>	1083
<u>8.67.39 mb_list_encodings() : Retourne un tableau contenant tous les encodages supportés</u>	1084
<u>8.67.40 mb_output_handler() : Fonction de traitement des affichages</u>	1085
<u>8.67.41 mb_parse_str() : Analyse les données HTTP GET/POST/COOKIE et assigne les variables globales</u>	1086
<u>8.67.42 mb_preferred_mime_name() : Détecte l'encodage MIME</u>	1086
<u>8.67.43 mb_regex_encoding() : Retourne le jeu de caractères courant pour les expressions rationnelles</u>	1087
<u>8.67.44 mb_regex_set_options() : Lit et modifie les options des fonctions d'expression rationnelle à support de caractères multi-octets</u>	1087
<u>8.67.45 mb_send_mail() : Envoie un mail encodé</u>	1087
<u>8.67.46 mb_split() : Scinde une chaîne en tableau avec une expression rationnelle multi-octets</u>	1088
<u>8.67.47 mb_strcut() : Coupe une partie de chaîne</u>	1088
<u>8.67.48 mb_strimwidth() : Tronque une chaîne</u>	1088
<u>8.67.49 mb_strlen() : Retourne la taille d'une chaîne</u>	1089
<u>8.67.50 mb_strpos() : Repère la première occurrence d'un caractère dans une chaîne</u>	1089
<u>8.67.51 mb_strrpos() : Repère la dernière occurrence d'un caractère dans une chaîne</u>	1089
<u>8.67.52 mb_strtolower() : Met tous les caractères en minuscules</u>	1090
<u>8.67.53 mb_strtoupper() : Met tous les caractères en majuscules</u>	1090
<u>8.67.54 mb_strwidth() : Retourne la largeur d'une chaîne</u>	1091
<u>8.67.55 mb_substitute_character() : Lit/modifie les caractères de substitution</u>	1091
<u>8.67.56 mb_substr_count() : Compte le nombre d'occurrences d'une sous-chaîne</u> ..	1092
<u>8.67.57 mb_substr() : Lit une sous-chaîne</u>	1092
8.68 MCAL	1093
<u>8.68.1 Introduction</u>	1093
<u>8.68.2 Pré-requis</u>	1093
<u>8.68.3 Installation</u>	1093
<u>8.68.4 Configuration à l'exécution</u>	1093
<u>8.68.5 Types de ressources</u>	1094
<u>8.68.6 Constantes pré-définies</u>	1094
<u>8.68.8 mcald_close() : Ferme une connexion MCAL</u>	1095

Sommaire

8 Référence des fonctions

<u>8.68.9 mcal_create_calendar()</u> : Crée un nouveau calendrier.....	1095
<u>8.68.10 mcal_date_compare()</u> : Compare deux dates.....	1096
<u>8.68.11 mcal_date_valid()</u> : Valide une date.....	1096
<u>8.68.12 mcal_day_of_week()</u> : Le jour de la semaine.....	1096
<u>8.68.13 mcal_day_of_year()</u> : Le jour de l'année.....	1096
<u>8.68.14 mcal_days_in_month()</u> : Retourne le nombre de jours dans un mois.....	1096
<u>8.68.15 mcal_delete_calendar()</u> : Efface un calendrier.....	1096
<u>8.68.16 mcal_delete_event()</u> : Efface un événement dans un calendrier MCAL.....	1097
<u>8.68.17 mcal_event_add_attribute()</u> : Ajoute un attribut et une valeur à la structure globale.....	1097
<u>8.68.18 mcal_event_init()</u> : Initialise la structure globale d'un flux MCAL.....	1097
<u>8.68.19 mcal_event_set_alarm()</u> : Fixe l'alarme de la structure globale.....	1097
<u>8.68.20 mcal_event_set_category()</u> : Fixe la catégorie de la structure globale.....	1097
<u>8.68.21 mcal_event_set_class()</u> : Fixe la classe de la structure globale.....	1097
<u>8.68.22 mcal_event_set_description()</u> : Fixe la description de la structure globale.....	1098
<u>8.68.23 mcal_event_set_end()</u> : Fixe la date de fin de la structure globale.....	1098
<u>8.68.24 mcal_event_set_recur_daily()</u> : Fixe la récurrence quotidienne.....	1098
<u>8.68.25 mcal_event_set_recur_monthly_mday()</u> : Fixe la récurrence.....	1098
<u>8.68.26 mcal_event_set_recur_monthly_wday()</u> : Fixe la récurrence mensuelle.....	1098
<u>8.68.27 mcal_event_set_recur_none()</u> : Supprime la récurrence de la structure globale.....	1098
<u>8.68.28 mcal_event_set_recur_weekly()</u> : Fixe la récurrence hebdomadaire.....	1099
<u>8.68.29 mcal_event_set_recur_yearly()</u> : Fixe la récurrence annuelle.....	1099
<u>8.68.30 mcal_event_set_start()</u> : Fixe la date de début de la structure globale.....	1099
<u>8.68.31 mcal_event_set_title()</u> : Fixe le titre de la structure globale.....	1099
<u>8.68.32 mcal_expunge()</u> : Supprime tous les événements marqués pour l'effacement.....	1099
<u>8.68.33 mcal_fetch_current_stream_event()</u> : Retourne un objet contenant la structure de date.....	1099
<u>8.68.34 mcal_fetch_event()</u> : Recherche un événement dans le calendrier.....	1100
<u>8.68.35 mcal_is_leap_year()</u> : Vérifie que l'année est bissextile.....	1101
<u>8.68.36 mcal_list_alarms()</u> : Retourne une liste d'événements qui ont une alarme prévue à une date.....	1101
<u>8.68.37 mcal_list_events()</u> : Retourne une liste d'événements pour une date ou entre deux dates.....	1101
<u>8.68.38 mcal_next_recurrence()</u> : Retourne la prochaine occurrence d'un événement.....	1101
<u>8.68.39 mcal_open()</u> : Ouvre une connexion MCAL.....	1102
<u>8.68.40 mcal_popen()</u> : Ouvre une connexion persistante MCAL.....	1102
<u>8.68.41 mcal_rename_calendar()</u> : Renomme un calendrier.....	1102
<u>8.68.42 mcal_reopen()</u> : Rouvre une connexion MCAL.....	1102
<u>8.68.43 mcal_snooze()</u> : Eteind l'alarme d'un événement.....	1102
<u>8.68.44 mcal_store_event()</u> : Modifie un événement dans un calendrier MCAL.....	1102
<u>8.68.45 mcal_time_valid()</u> : Valide une heure.....	1103
<u>8.68.46 mcal_week_of_year()</u> : Retourne le numéro de semaine dans l'année pour une date donnée.....	1103
<u>8.69 chiffrement mcrypt</u>	1104
<u>8.69.1 Introduction</u>	1104
<u>8.69.2 Pré-requis</u>	1104
<u>8.69.3 Installation</u>	1104
<u>8.69.4 Configuration à l'exécution</u>	1104

Sommaire

8 Référence des fonctions

<u>8.69.5 Types de ressources</u>	1104
<u>8.69.6 Constantes pré-définies</u>	1104
<u>8.69.7 Mcrypt ciphers</u>	1105
<u>8.69.8 Exemples</u>	1106
<u>8.69.10 mcrypt_cfb() : Chiffre/déchiffre des données en mode CFB</u>	1108
<u>8.69.11 mcrypt_create_iv() : Crée un vecteur d'initialisation à partir d'une source aléatoire</u>	1108
<u>8.69.12 mcrypt_decrypt() : Déchiffre un texte avec les paramètres donnés</u>	1109
<u>8.69.13 mcrypt_ecb() : Obsolète : Chiffre/déchiffre des données en mode ECB</u>	1109
<u>8.69.14 mcrypt_enc_get_algorithms_name() : Retourne le nom de l'algorithme</u>	1109
<u>8.69.15 mcrypt_enc_get_block_size() : Retourne la taille de blocs d'un algorithme</u>	1110
<u>8.69.16 mcrypt_enc_get_iv_size() : Retourne la taille du VI d'un algorithme</u>	1110
<u>8.69.17 mcrypt_enc_get_key_size() : Retourne la taille maximale de la clé pour un mode</u>	1110
<u>8.69.18 mcrypt_enc_get_modes_name() : Retourne le nom du mode</u>	1110
<u>8.69.19 mcrypt_enc_get_supported_key_sizes() : Retourne un tableau contenant les tailles de clés acceptées par un algorithme</u>	1111
<u>8.69.20 mcrypt_enc_is_block_algorithm_mode() : Teste le chiffrement par blocs d'un mode</u>	1111
<u>8.69.21 mcrypt_enc_is_block_algorithm() : Teste le chiffrement par blocs d'un algorithme</u>	1111
<u>8.69.22 mcrypt_enc_is_block_mode() : Teste si le mode retourne les données par blocs</u>	1111
<u>8.69.23 mcrypt_enc_self_test() : Teste un module ouvert</u>	1112
<u>8.69.24 mcrypt_encrypt() : Chiffre un texte</u>	1112
<u>8.69.25 mcrypt_generic_deinit() : Prépare le module pour le déchargement</u>	1113
<u>8.69.26 mcrypt_generic_end() : Termine un chiffrement</u>	1113
<u>8.69.27 mcrypt_generic_init() : Initialise tous les buffers nécessaires</u>	1113
<u>8.69.28 mcrypt_generic() : Chiffre les données</u>	1113
<u>8.69.29 mcrypt_get_block_size() : Retourne la taille de blocs d'un chiffrement</u>	1114
<u>8.69.30 mcrypt_get_cipher_name() : Lit le nom du chiffrement utilisé</u>	1115
<u>8.69.31 mcrypt_get_iv_size() : Retourne la taille du VI utilisé par un couple chiffrement/mode</u>	1115
<u>8.69.32 mcrypt_get_key_size() : Retourne la taille de la clé d'un chiffrement</u>	1115
<u>8.69.33 mcrypt_list_algorithms() : Liste tous les algorithmes de chiffrement supportés</u>	1116
<u>8.69.34 mcrypt_list_modes() : Liste tous les modes de chiffrement supportés</u>	1116
<u>8.69.35 mcrypt_module_close() : Décharge le module de chiffrement</u>	1117
<u>8.69.36 mcrypt_module_get_algo_block_size() : Retourne la taille de blocs d'un algorithme</u>	1117
<u>8.69.37 mcrypt_module_get_algo_key_size() : Retourne la taille maximale de clé</u>	1117
<u>8.69.38 mcrypt_module_get_supported_key_sizes() : Retourne un tableau contenant les tailles de clés supportées par un algorithme de chiffrement</u>	1117
<u>8.69.39 mcrypt_module_is_block_algorithm_mode() : Indique si un mode fonctionne par blocs</u>	1118
<u>8.69.40 mcrypt_module_is_block_algorithm() : Indique si un algorithme fonctionne par blocs</u>	1118
<u>8.69.41 mcrypt_module_is_block_mode() : Indique si un mode travaille par blocs</u>	1118
<u>8.69.42 mcrypt_module_open() : Ouvre le module de l'algorithme et du mode à utiliser</u>	1118

Sommaire

8 Référence des fonctions

<u>8.69.43 mcrypt_module_self_test()</u> : Teste un mode.....	1119
<u>8.69.44 mcrypt_ofb()</u> : Chiffre/déchiffre des données en mode OFB.....	1120
<u>8.69.45 mdcrypt_generic()</u> : Déchiffre.....	1120
8.70 Fonctions de paiement MCVE (Monetra)	1122
<u>8.70.1 Introduction</u>	1122
<u>8.70.2 Installation</u>	1122
<u>8.70.3 Voir aussi</u>	1123
<u>8.70.4 Constantes pré-définies</u>	1123
<u>8.70.6 m_completeauthorizations()</u> : Nombre d'autorisations complètes en attente, retourne un tableau de leur identifiant.....	1124
<u>8.70.7 m_connect()</u> : Établit une connexion à MCVE.....	1124
<u>8.70.8 m_connectionerror()</u> : Obtient une représentation textuelle de la cause de l'échec de connexion.....	1124
<u>8.70.9 m_deletetrans()</u> : Supprime une transaction spécifiée de la structure MCVE_CONN.....	1124
<u>8.70.10 m_destroyconn()</u> : Détruit la connexion et la structure MCVE_CONN.....	1124
<u>8.70.11 m_destroyengine()</u> : Libère la mémoire associée avec la connectivité IP/SSL.....	1125
<u>8.70.12 m_getcell()</u> : Récupère une cellule spécifique d'une réponse délimitée par virgule avec le nom de colonne.....	1125
<u>8.70.13 m_getcellbynum()</u> : Récupère une cellule spécifique d'une réponse délimitée par virgule avec le numéro de colonne.....	1125
<u>8.70.14 m_getcommadelimited()</u> : Récupère les données brutes délimitées par virgule retournées par MCVE.....	1125
<u>8.70.15 m_getheader()</u> : Récupère le nom de la colonne dans la réponse délimitée par virgule.....	1125
<u>8.70.16 m_initconn()</u> : Crée et initialise une structure MCVE_CONN.....	1125
<u>8.70.17 m_initengine()</u> : Apprête le client pour Communication IP/SSL.....	1126
<u>8.70.18 m_iscommadelimited()</u> : Vérifie si la réponse est délimitée par virgule.....	1126
<u>8.70.19 m_maxconntimeout()</u> : Le temps maximal que l'API tentera une connexion à MCVE.....	1126
<u>8.70.20 m_monitor()</u> : Effectue une communication avec MCVE (envoi/réception de données) en mode non-bloquant.....	1126
<u>8.70.21 m_numcolumns()</u> : Nombre de colonnes retournées dans une réponse délimitée par virgule.....	1126
<u>8.70.22 m_numrows()</u> : Nombre de lignes retournés dans une réponse délimitée par virgule.....	1126
<u>8.70.23 m_parsecommadelimited()</u> : Analyse la réponse délimitée par virgule alors m_getcell, etc. fonctionnera.....	1127
<u>8.70.24 m_responsekeys()</u> : Retourne un tableau de chaînes de caractères qui représente les clés qui peuvent être utilisées en tant que paramètres de réponse sur cette transaction.....	1127
<u>8.70.25 m_responseparam()</u> : Récupère un paramètre spécial de réponse.....	1127
<u>8.70.26 m_returnstatus()</u> : Vérifie si la transaction fut complétée correctement.....	1127
<u>8.70.27 m_setblocking()</u> : Fixe le mode bloquant/non-bloquant pour une connexion.....	1127
<u>8.70.28 m_setdropfile()</u> : Fixe la méthode de connexion à Drop-File.....	1127
<u>8.70.29 m_setip()</u> : Fixe la méthode de connexion à IP.....	1128
<u>8.70.30 m_setssl_cafile()</u> : Fixe le fichier SSL CA (Certificate Authority) pour vérification du certificat du serveur.....	1128
<u>8.70.31 m_setssl_files()</u> : Fixe les fichiers de clé de certificat et certifie si le	

Sommaire

8 Référence des fonctions

<u>serveur requiert la vérification du client par certificat</u>	1128
<u>8.70.32 m_setssl()</u> : Fixe la méthode de connexion à SSL.....	1128
<u>8.70.33 m_settimeout()</u> : Fixe le temps maximal d'une transaction (par transaction).....	1128
<u>8.70.34 m_sslcert_gen_hash()</u> : Génère une clé de hachage pour les vérifications de certificat de client SSL.....	1128
<u>8.70.35 m_transactionsent()</u> : Vérifie si le tampon de sortie est vide.....	1129
<u>8.70.36 m_transinqueue()</u> : Nombre de transactions dans la file de client.....	1129
<u>8.70.37 m_transkeyval()</u> : Ajoute une paire clé/valeur d'une transaction. Remplace <u>transparam()</u> dépréciée.....	1129
<u>8.70.38 m_transnew()</u> : Démarre une nouvelle transaction.....	1129
<u>8.70.39 m_transsend()</u> : Finalise et envoie la transaction.....	1129
<u>8.70.40 m_uwait()</u> : Attend x micro secondes.....	1129
<u>8.70.41 m_validateidentif()</u> : Valide ou non l'identifiant passé sur n'importe quelle transaction qu'il est passé.....	1129
<u>8.70.42 m_verifyconnection()</u> : Ping ou non sur la connexion pour la vérifier.....	1130
<u>8.70.43 m_verifysslcert()</u> : Vérifie ou non le certificat du serveur SSL.....	1130
<u>8.71 Fonctions Memcache</u>	1131
<u>8.71.1 Introduction</u>	1131
<u>8.71.2 Pré-requis</u>	1131
<u>8.71.3 Installation</u>	1131
<u>8.71.4 Constantes pré-définies</u>	1131
<u>8.71.5 Configuration à l'exécution</u>	1131
<u>8.71.6 Types de ressources</u>	1132
<u>8.71.7 Exemples</u>	1132
<u>8.71.9 Memcache::close()</u> : Ferme la connexion avec le serveur Memcache.....	1133
<u>8.71.10 Memcache::connect()</u> : Ouvre une connexion avec le serveur Memcache.....	1133
<u>8.71.11 memcache_debug()</u> : Active ou non l'affichage des informations de débogage.....	1134
<u>8.71.12 Memcache::decrement()</u> : Décrémente la valeur d'un élément.....	1134
<u>8.71.13 Memcache::delete()</u> : Efface un élément du serveur de cache.....	1135
<u>8.71.14 Memcache::flush()</u> : Efface tous les éléments existant sur le serveur de cache.....	1135
<u>8.71.15 Memcache::get()</u> : Récupère un élément du serveur de cache.....	1136
<u>8.71.16 Memcache::getStats()</u> : Lit les statistiques du serveur.....	1136
<u>8.71.17 Memcache::getVersion()</u> : Retourne le numéro de version du serveur.....	1137
<u>8.71.18 Memcache::increment()</u> : Incrément la valeur d'un élément.....	1137
<u>8.71.19 Memcache::pconnect()</u> : Ouvre une connexion persistante à un serveur de cache.....	1138
<u>8.71.20 Memcache::replace()</u> : Remplace une valeur d'un élément existant.....	1138
<u>8.71.21 Memcache::set()</u> : Stocke des données dans le serveur de cache.....	1139
<u>8.72 Hash</u>	1141
<u>8.72.1 Introduction</u>	1141
<u>8.72.2 Pré-requis</u>	1141
<u>8.72.3 Installation</u>	1141
<u>8.72.4 Configuration à l'exécution</u>	1141
<u>8.72.5 Types de ressources</u>	1141
<u>8.72.6 Constantes pré-définies</u>	1141
<u>8.72.7 Exemples</u>	1142
<u>8.72.9 mhash_get_block_size()</u> : Retourne la taille de bloc du hash.....	1142
<u>8.72.10 mhash_get_hash_name()</u> : Retourne le nom du hash.....	1143

Sommaire

8 Référence des fonctions

<u>8.72.11 mhash_keygen_s2k()</u> : Génère une clé.....	1143
<u>8.72.12 mhash()</u> : Calcule un hash.....	1143
8.73 Ming pour Flash	1144
<u>8.73.1 Introduction</u>	1144
<u>8.73.2 Pré-requis</u>	1144
<u>8.73.3 Installation</u>	1144
<u>8.73.4 Configuration à l'exécution</u>	1145
<u>8.73.5 Types de ressources</u>	1145
<u>8.73.6 Constantes pré-définies</u>	1145
<u>8.73.7 Classes pré-définies</u>	1146
<u>8.73.9 ming_setcubicthreshold()</u> : Fixe le niveau de seuil cubique (??).....	1149
<u>8.73.10 ming_setscale()</u> : Configure l'échelle (??).....	1149
<u>8.73.11 ming_useConstants()</u> : Utilise une constante de réserve (?).....	1149
<u>8.73.12 ming_useswfversion()</u> : Utilise la version de SWF (??).....	1149
<u>8.73.13 SWFAction()</u> : Crée une nouvelle action.....	1149
<u>8.73.14 SWFBitmap->getHeight()</u> : Retourne la hauteur d'une bitmap Flash.....	1156
<u>8.73.15 SWFBitmap->getWidth()</u> : Retourne la largeur d'une bitmap Flash.....	1156
<u>8.73.16 SWFBitmap()</u> : Crée un objet bitmap.....	1157
<u>8.73.17 SWFbutton->addAction()</u> : Ajoute une action au bouton Flash.....	1158
<u>8.73.18 SWFButton::addASound()</u> : Associe un son avec une transition de bouton.....	1158
<u>8.73.19 SWFbutton->addShape()</u> : Ajoute une forme à un bouton.....	1159
<u>8.73.20 SWFbutton->setAction()</u> : Assigne l'action du bouton.....	1159
<u>8.73.21 SWFbutton->setdown()</u> : Alias de addShape(shape, SWFBUTTON_DOWN).....	1159
<u>8.73.22 SWFbutton->setHit()</u> : Alias de addShape(shape, SWFBUTTON_HIT).....	1159
<u>8.73.23 SWFButton::setMenu()</u> : Active le traçage comme comportement de menu de bouton.....	1160
<u>8.73.24 SWFbutton->setOver()</u> : Alias de addShape(shape, SWFBUTTON_OVER).....	1160
<u>8.73.25 SWFbutton->setUp()</u> : Alias de SWFbutton->addShape(shape, SWFBUTTON_UP).....	1160
<u>8.73.26 SWFbutton()</u> : Crée un nouveau bouton.....	1160
<u>8.73.27 SWFDisplayItem::addAction()</u> : Ajoute cet SWFAction à l'instance donnée SWFSprite.....	1162
<u>8.73.28 SWFDisplayItem->addColor()</u> : Ajoute une couleur à une transformation.....	1162
<u>8.73.29 SWFDisplayItem::endMask()</u> : Un autre moyen pour définir une couche MASK.....	1163
<u>8.73.30 SWFDisplayItem::getRot()</u> :	1163
<u>8.73.31 SWFDisplayItem::getX()</u> :	1163
<u>8.73.32 SWFDisplayItem::getXScale()</u> :	1163
<u>8.73.33 SWFDisplayItem::getXSkew()</u> :	1163
<u>8.73.34 SWFDisplayItem::getY()</u> :	1163
<u>8.73.35 SWFDisplayItem::getYScale()</u> :	1164
<u>8.73.36 SWFDisplayItem::getYSkew()</u> :	1164
<u>8.73.37 SWFDisplayItem->move()</u> : Déplace un objet en coordonnées relatives.....	1164
<u>8.73.38 SWFDisplayItem->moveTo()</u> : Déplace un objet en coordonnées globales.....	1164
<u>8.73.39 SWFDisplayItem->multColor()</u> : Multiplie la couleur de transformation.....	1165
<u>8.73.40 SWFDisplayItem->remove()</u> : Supprime un objet d'une animation.....	1165
<u>8.73.41 SWFDisplayItem->Rotate()</u> : Fait tourner une forme relativement.....	1166

Sommaire

8 Référence des fonctions

<u>8.73.42 SWFDisplayItem->rotateTo()</u> : Tourne un objet en angle absolu.....	1166
<u>8.73.43 SWFDisplayItem->scale()</u> : Etire un objet relativement.....	1167
<u>8.73.44 SWFDisplayItem->scaleTo()</u> : Etire un objet en coordonnées globales.....	1167
<u>8.73.45 SWFDisplayItem->setDepth()</u> : Modifie la place en profondeur (z-order).....	1168
<u>8.73.46 SWFDisplayItem::setMaskLevel()</u> : Définit une couche MASK à un niveau.....	1168
<u>8.73.47 SWFDisplayItem::setMatrix()</u> : Fixe la matrice de transformation de l'item.....	1168
<u>8.73.48 SWFDisplayItem->setName()</u> : Nomme un objet.....	1168
<u>8.73.49 SWFDisplayItem->setRatio()</u> : Modifie le ratio de l'objet.....	1169
<u>8.73.50 SWFDisplayItem->skewX()</u> : Incline suivant les X relativement.....	1170
<u>8.73.51 SWFDisplayItem->skewXTo()</u> : Incline suivant les X.....	1170
<u>8.73.52 SWFDisplayItem->skewY()</u> : Incline suivant les Y relativement.....	1170
<u>8.73.53 SWFDisplayItem->skewYTo()</u> : Incline suivant les Y.....	1171
<u>8.73.54 SWFFill->moveTo()</u> : Déplace l'origine de l'objet SWFFill.....	1171
<u>8.73.55 SWFFill->rotateTo()</u> : Tourne la forme.....	1171
<u>8.73.56 SWFFill->scaleTo()</u> : Modifie l'échelle de la forme.....	1172
<u>8.73.57 SWFFill->skewXTo()</u> : Incline (abscisses).....	1172
<u>8.73.58 SWFFill->skewYTo()</u> : Incline (ordonnées).....	1172
<u>8.73.59 SWFFill()</u> : Crée un objet de remplissage.....	1172
<u>8.73.60 SWFFont::getAscent()</u> : Retourne la hampe de la police ou 0 si elle est non disponible.....	1173
<u>8.73.61 SWFFont::getDescent()</u> : Retourne le jambage de la police ou 0 s'il n'est pas disponible.....	1173
<u>8.73.62 SWFFont::getLeading()</u> : Retourne l'interlignage de la police ou 0 s'il n'est pas disponible.....	1173
<u>8.73.63 SWFFont::getShape()</u> : Retourne la forme glyphe d'un caractère en tant que chaîne de caractères.....	1173
<u>8.73.64 SWFFont::getUTF8Width()</u> : Calcule la largeur de la chaîne de caractères donnée dans cette police à ca pleine taille.....	1173
<u>8.73.65 swffont->getwidth()</u> : Retourne la taille de la chaîne.....	1173
<u>8.73.66 SWFFont()</u> : Charge une police.....	1174
<u>8.73.67 SWFFontChar::addChars()</u> : Ajoute des caractères à une police pour exporter une police.....	1174
<u>8.73.68 SWFFontChar::addUTF8Chars()</u> : Ajoute des caractères à une police pour exporter une police.....	1174
<u>8.73.69 SWFGradient->addEntry()</u> : Ajoute une couleur à la liste du gradient.....	1175
<u>8.73.70 SWFGradient()</u> : Crée un objet gradient.....	1175
<u>8.73.71 SWFMorph->getshape1()</u> : Sélectionne la forme de départ.....	1176
<u>8.73.72 SWFMorph->getshape2()</u> : Sélectionne la forme de fin.....	1176
<u>8.73.73 SWFMorph()</u> : Crée un morphing.....	1176
<u>8.73.74 SWFMovie->add()</u> : Ajoute un objet dans une animation.....	1177
<u>8.73.75 SWFMovie::addExport()</u> :.....	1178
<u>8.73.76 SWFMovie::addFont()</u> :.....	1178
<u>8.73.77 SWFMovie::importChar()</u> :.....	1178
<u>8.73.78 SWFMovie::importFont()</u> :.....	1178
<u>8.73.79 SWFMovie::labelFrame()</u> : Écrit un cadre.....	1178
<u>8.73.80 SWFMovie->nextframe()</u> : Passe à l'image suivante.....	1178
<u>8.73.81 SWFMovie->output()</u> : Envoie votre animation au navigateur.....	1179
<u>8.73.82 swfmovie->remove()</u> : Supprime un objet d'une animation.....	1179
<u>8.73.83 SWFMovie->save()</u> : Sauve dans un fichier.....	1179
<u>8.73.84 SWFMovie::saveToFile()</u> :.....	1180

Sommaire

8 Référence des fonctions

<u>8.73.85 SWFMovie->setbackground() : Modifie la couleur de fond.....</u>	1180
<u>8.73.86 SWFMovie->setdimension() : Modifie les dimensions de l'animation.....</u>	1180
<u>8.73.87 SWFMovie->setframes() : Modifie le nombre total d'images dans l'animation.....</u>	1180
<u>8.73.88 SWFMovie->setrate() : Modifie la vitesse de l'animation.....</u>	1181
<u>8.73.89 SWFMovie::startSound() :</u>	1181
<u>8.73.90 SWFMovie::stopSound() :</u>	1181
<u>8.73.91 SWFMovie->streammp3() : Envoie un fichier MP3 en streaming.....</u>	1181
<u>8.73.92 SWFMovie::writeExports() :</u>	1182
<u>8.73.93 SWFMovie() : Crée un objet 'animation'.....</u>	1182
<u>8.73.94 SWFPrebuiltClip() : Retourne un objet SWFPrebuiltClip.....</u>	1182
<u>8.73.95 SWFShape->addFill() : Ajoute un remplissage plein à la forme Flash.....</u>	1182
<u>8.73.96 SWFShape::drawArc() : Dessine un arc de radius r centré à la position courante, d'un angle startAngle à endAngle mesuré dans le sens des aiguilles d'une montre à partir de midi.....</u>	1184
<u>8.73.97 SWFShape::drawCircle() : Dessine un cercle de radius r centré à la position courante, dans le sens inverse des aiguilles d'une montre.....</u>	1184
<u>8.73.98 SWFShape::drawCubic() : Dessine une courbe de Bézier cubique en utilisant la position courante et les trois points donnés en tant que position de contrôle.....</u>	1184
<u>8.73.99 SWFShape::drawCubicTo() : Dessine une courbe de Bézier cubique en utilisant la position courante et les trois points donnés en tant que points de contrôle.....</u>	1184
<u>8.73.100 SWFShape->drawCurve() : Dessine une courbe relativement.....</u>	1184
<u>8.73.101 SWFShape->drawCurveTo() : Dessine une courbe.....</u>	1185
<u>8.73.102 SWFShape::drawGlyph() : Dessine le premier caractère dans la chaîne de caractères donnée dans la forme en utilisant la définition de glyphe de la police donnée.....</u>	1185
<u>8.73.103 SWFShape->drawLine() : Dessine une ligne relativement.....</u>	1185
<u>8.73.104 SWFShape->drawLineTo() : Dessine une ligne.....</u>	1186
<u>8.73.105 SWFShape->movePen() : Déplace le stylo relativement.....</u>	1186
<u>8.73.106 SWFShape->movePenTo() : Déplace le stylo.....</u>	1186
<u>8.73.107 SWFShape->setLeftFill() : Modifie la couleur de rastérisation de gauche.....</u>	1187
<u>8.73.108 SWFShape->setLine() : Modifie le style de ligne de la forme.....</u>	1187
<u>8.73.109 SWFShape->setRightFill() : Modifie la couleur de rastérisation de droite.....</u>	1188
<u>8.73.110 SWFShape() : Crée une nouvelle forme.....</u>	1189
<u>8.73.111 SWFSound() : Retourne un nouvel objet SWFSound d'un fichier donné.....</u>	1189
<u>8.73.112 SWFSoundInstance::loopCount() :</u>	1190
<u>8.73.113 SWFSoundInstance::loopInPoint() :</u>	1190
<u>8.73.114 SWFSoundInstance::loopOutPoint() :</u>	1190
<u>8.73.115 SWFSoundInstance::noMultiple() :</u>	1190
<u>8.73.116 swfsprite->add() : Ajoute un objet à un sprite.....</u>	1190
<u>8.73.117 SWFSprite::labelFrame() : Écrit un cadre.....</u>	1190
<u>8.73.118 SWFSprite->nextframe() : Va à la prochaine image du sprite.....</u>	1191
<u>8.73.119 SWFSprite->remove() : Supprime un objet dans un sprite.....</u>	1191
<u>8.73.120 SWFSprite->setframes() : Fixe le nombre maximum d'images dans le sprite.....</u>	1191
<u>8.73.121 SWFSprite::startSound() :</u>	1191
<u>8.73.122 SWFSprite::stopSound() :</u>	1191
<u>8.73.123 SWFSprite() : Crée un sprite.....</u>	1192
<u>8.73.124 SWFText->addString() : Ajoute du texte.....</u>	1192

Sommaire

8 Référence des fonctions

<u>8.73.125 SWFText::addUTF8String()</u> : Écrit le texte donné dans l'objet SWFText à la position du crayon, en utilisant la police, la hauteur, l'espacement et la couleur courante.....	1193
<u>8.73.126 SWFText::getAscent()</u> : Retourne la hampe de la police courante à sa taille courante ou 0 si elle n'est pas disponible.....	1193
<u>8.73.127 SWFText::getDescent()</u> : Retourne le jambage de la police courante à sa taille courante ou 0 s'il n'est pas disponible.....	1193
<u>8.73.128 SWFText::getLeading()</u> : Retourne l'interlignage de la police courante à sa taille courante ou 0 s'il n'est pas disponible.....	1193
<u>8.73.129 SWFText::getUTF8Width()</u> : Calcule la largeur de la chaîne de caractères donnée dans les objets de texte de police et taille courante.....	1193
<u>8.73.130 SWFText->getWidth()</u> : Calcule la longueur d'une chaîne.....	1193
<u>8.73.131 SWFText->moveTo()</u> : Déplace le stylo de texte.....	1194
<u>8.73.132 SWFText->setColor()</u> : Modifie la couleur de la police.....	1194
<u>8.73.133 SWFText->setFont()</u> : Sélectionne la police courante.....	1194
<u>8.73.134 SWFText->setHeight()</u> : Modifie la hauteur de la police courante.....	1194
<u>8.73.135 SWFText->setSpacing()</u> : Modifie l'espacement de police.....	1195
<u>8.73.136 SWFText()</u> : Crée un nouvel objet texte.....	1195
<u>8.73.137 SWFTextField::addChars()</u> : Ajoute des caractères à une police qui seront disponibles dans un champ texte.....	1195
<u>8.73.138 SWFTextField->addstring()</u> : Ajoute au texte.....	1196
<u>8.73.139 SWFTextField->align()</u> : Modifie l'alignement du texte.....	1196
<u>8.73.140 SWFTextField->setbounds()</u> : Sélectionne la largeur et hauteur du champ.....	1196
<u>8.73.141 SWFTextField->setcolor()</u> : Modifie la couleur du champ texte.....	1196
<u>8.73.142 SWFTextField->setFont()</u> : Modifie la police du champ.....	1197
<u>8.73.143 SWFTextField->setHeight()</u> : Modifie la hauteur de la police du champ texte.....	1197
<u>8.73.144 SWFTextField->setindentation()</u> : Modifie l'indentation de la première ligne.....	1197
<u>8.73.145 SWFTextField->setLeftMargin()</u> : Modifie la marge de gauche.....	1197
<u>8.73.146 SWFTextField->setLineSpacing()</u> : Modifie l'espacement de lignes.....	1198
<u>8.73.147 SWFTextField->setMargins()</u> : Modifie les marges du champ texte Flash.....	1198
<u>8.73.148 SWFTextField->setname()</u> : Nomme le champ texte.....	1198
<u>8.73.149 SWFTextField::setPadding()</u> : Fixe le remplissage d'un champ de texte.....	1198
<u>8.73.150 SWFTextField->setrightMargin()</u> : Modifie la marge de droite.....	1198
<u>8.73.151 SWFTextField()</u> : Crée un nouveau champ texte.....	1199
<u>8.73.152 SWFVideoStream::getNumFrames()</u> :	1199
<u>8.73.153 SWFVideoStream::setDimension()</u> :	1200
<u>8.73.154 SWFVideoStream()</u> : Retourne un objet SWFVideoStream.....	1200
<u>8.74 Fonctions diverses</u>	1201
<u>8.74.1 Introduction</u>	1201
<u>8.74.2 Pré-requis</u>	1201
<u>8.74.3 Installation</u>	1201
<u>8.74.4 Configuration à l'exécution</u>	1201
<u>8.74.5 Types de ressources</u>	1201
<u>8.74.6 Constantes pré-définies</u>	1202
<u>8.74.8 connection_status()</u> : Retourne les bits de statut de la connexion HTTP.....	1202
<u>8.74.9 connection_timeout()</u> : Indique si le script a expiré.....	1203
<u>8.74.10 constant()</u> : Retourne la valeur d'une constante.....	1203

Sommaire

8 Référence des fonctions

<u>8.74.11 define()</u> : Définit une constante.....	1203
<u>8.74.12 defined()</u> : Vérifie l'existence d'une constante.....	1204
<u>8.74.13 die()</u> : Alias de la fonction exit.....	1204
<u>8.74.14 eval()</u> : Exécute une chaîne comme un script PHP.....	1204
<u>8.74.15 exit()</u> : Termine le script courant.....	1205
<u>8.74.16 get_browser()</u> : Indique de quoi est capable le navigateur client.....	1206
<u>8.74.17 halt_compiler()</u> : Stoppe l'exécution du compilateur.....	1207
<u>8.74.18 highlight_file()</u> : Colorisation syntaxique d'un fichier.....	1208
<u>8.74.19 highlight_string()</u> : Applique la syntaxe colorisée à du code PHP.....	1208
<u>8.74.20 ignore_user_abort()</u> : Active l'interruption de script sur déconnexion du visiteur.....	1209
<u>8.74.21 pack()</u> : Compacte des données dans une chaîne binaire.....	1209
<u>8.74.22 php_check_syntax()</u> : Vérifie la syntaxe PHP (et exécute) d'un fichier spécifique.....	1210
<u>8.74.23 php_strip_whitespace()</u> : Retourne la source sans commentaires ni espaces blancs.....	1211
<u>8.74.24 show_source()</u> : Alias de highlight_file.....	1211
<u>8.74.25 sleep()</u> : Arrête l'exécution durant quelques secondes.....	1211
<u>8.74.26 time_nanosleep()</u> : Attendre pendant un nombre de secondes et de nanosecondes.....	1211
<u>8.74.27 time_sleep_until()</u> : Stoppe le script pendant une durée spécifiée.....	1212
<u>8.74.28 uniqid()</u> : Génère un identifiant unique.....	1212
<u>8.74.29 unpack()</u> : Déconditionne des données depuis une chaîne binaire.....	1212
<u>8.74.30 usleep()</u> : Retarde l'exécution en micro-secondes.....	1213
<u>8.75 mnoGoSearch</u>	1214
<u>8.75.1 Introduction</u>	1214
<u>8.75.2 Pré-requis</u>	1214
<u>8.75.3 Installation</u>	1214
<u>8.75.4 Configuration à l'exécution</u>	1214
<u>8.75.5 Types de ressources</u>	1215
<u>8.75.6 Constantes pré-définies</u>	1215
<u>8.75.8 udm_alloc_agent_array()</u> : Alloue une session mnoGoSearch.....	1217
<u>8.75.9 udm_alloc_agent()</u> : Alloue une session mnoGoSearch.....	1217
<u>8.75.10 udm_api_version()</u> : Lit la version de l'API mnoGoSearch.....	1218
<u>8.75.11 udm_cat_list()</u> : Liste toutes les catégories soeurs d'une catégorie.....	1218
<u>8.75.12 udm_cat_path()</u> : Lit le chemin de la catégorie courante.....	1219
<u>8.75.13 udm_check_charset()</u> : Vérifie si un jeu de caractères est supporté par mnogosearch.....	1220
<u>8.75.14 udm_check_stored()</u> : Vérifie la connexion MnoGoSearch avec le document stocké.....	1220
<u>8.75.15 udm_clear_search_limits()</u> : Annule toutes les limitations de recherche MnoGoSearch.....	1220
<u>8.75.16 udm_close_stored()</u> : Ferme la connexion MnoGoSearch avec le document enregistré.....	1220
<u>8.75.17 udm_crc32()</u> : Retourne la somme de contrôle CRC32 d'une chaîne.....	1221
<u>8.75.18 udm_errno()</u> : Numéro d'erreur mnoGoSearch.....	1221
<u>8.75.19 udm_error()</u> : Message d'erreur mnoGoSearch.....	1221
<u>8.75.20 udm_find()</u> : Effectue une recherche.....	1221
<u>8.75.21 udm_free_agent()</u> : Détruit une session mnoGoSearch.....	1222
<u>8.75.22 udm_free_ispell_data()</u> : Libère la mémoire allouée pour ispell par MnoGoSearch.....	1222

Sommaire

8 Référence des fonctions

<u>8.75.23 udm_free_res()</u> : Libère un résultat mnoGoSearch.....	1222
<u>8.75.24 udm_get_doc_count()</u> : Lit le nombre total de documents dans les bases MnoGoSearch.....	1222
<u>8.75.25 udm_get_res_field()</u> : Lit un champ de résultat mnoGoSearch.....	1223
<u>8.75.26 udm_get_res_param()</u> : Lit les paramètres de résultats mnoGoSearch.....	1223
<u>8.75.27 udm_hash32()</u> : Retourne une somme de contrôle Hash32.....	1224
<u>8.75.28 udm_load_ispell_data()</u> : Charge les données ispell.....	1224
<u>8.75.29 udm_open_stored()</u> : Ouvre une connexion MnoGoSearch avec un document stocké.....	1226
<u>8.75.30 udm_set_agent_param()</u> : Modifie les paramètres de l'agent MnoGoSearch.....	1226
<u>8.76 Sessions Mohawk</u>	1230
<u>8.76.1 Introduction</u>	1230
<u>8.76.2 Pré-requis</u>	1230
<u>8.76.3 Installation</u>	1230
<u>8.76.4 Configuration à l'exécution</u>	1230
<u>8.76.5 Types de ressources</u>	1230
<u>8.76.6 Constantes pré-définies</u>	1230
<u>8.76.8 msession_count()</u> : Compte le nombre de sessions.....	1231
<u>8.76.9 msession_create()</u> : Crée une session.....	1231
<u>8.76.10 msession_destroy()</u> : Détruit la session.....	1231
<u>8.76.11 msession_disconnect()</u> : Ferme la connexion au serveur msession.....	1231
<u>8.76.12 msession_find()</u> : Trouve toutes les sessions avec un nom et une valeur....	1231
<u>8.76.13 msession_get_array()</u> : Lit un tableau de variables msession.....	1231
<u>8.76.14 msession_get_data()</u> : Lit les données de la session.....	1232
<u>8.76.15 msession_get()</u> : Lit une valeur dans une session.....	1232
<u>8.76.16 msession_inc()</u> : Incrémente une valeur dans une session.....	1232
<u>8.76.17 msession_list()</u> : Liste toutes les sessions.....	1232
<u>8.76.18 msession_listvar()</u> : Liste les sessions contenant une variable.....	1232
<u>8.76.19 msession_lock()</u> : Verrouille une session.....	1232
<u>8.76.20 msession_plugin()</u> : Appele une fonction d'échappement dans les plugins de personnalité msession.....	1232
<u>8.76.21 msession_randstr()</u> : Lit une chaîne aléatoire.....	1233
<u>8.76.22 msession_set_array()</u> : Définit les variables msession depuis un tableau....	1233
<u>8.76.23 msession_set_data()</u> : Définit les données de session.....	1233
<u>8.76.24 msession_set()</u> : Modifie une valeur dans la session.....	1233
<u>8.76.25 msession_timeout()</u> : Lit et/ou modifie la durée de vie de la session.....	1233
<u>8.76.26 msession_uniq()</u> : Récupère un identifiant unique.....	1233
<u>8.76.27 msession_unlock()</u> : Déverrouille une session.....	1234
<u>8.77 mSQL</u>	1235
<u>8.77.1 Introduction</u>	1235
<u>8.77.2 Installation</u>	1235
<u>8.77.3 Configuration à l'exécution</u>	1235
<u>8.77.4 Types de ressources</u>	1235
<u>8.77.5 Constantes pré-définies</u>	1235
<u>8.77.6 Exemples</u>	1236
<u>8.77.8 msql_close()</u> : Ferme une connexion mSQL.....	1237
<u>8.77.9 msql_connect()</u> : Ouvre une connexion mSQL.....	1237
<u>8.77.10 msql_create_db()</u> : Crée une base de données mSQL.....	1238
<u>8.77.11 msql_createdb()</u> : Alias de msql_create_db.....	1238
<u>8.77.12 msql_data_seek()</u> : Déplace le pointeur interne mSQL.....	1238

Sommaire

8 Référence des fonctions

<u>8.77.13</u> <u>mysql_db_query()</u> : Soumet une requête mSQL.....	1238
<u>8.77.14</u> <u>mysql_dbname()</u> : Alias de <u>mysql_result</u>	1238
<u>8.77.15</u> <u>mysql_drop_db()</u> : Efface une base de données mSQL.....	1238
<u>8.77.16</u> <u>mysql_error()</u> : Retourne le message d'erreur mSQL.....	1238
<u>8.77.17</u> <u>mysql_fetch_array()</u> : Lit une ligne sous la forme d'un tableau.....	1239
<u>8.77.18</u> <u>mysql_fetch_field()</u> : Lit la valeur d'un champ mSQL.....	1239
<u>8.77.19</u> <u>mysql_fetch_object()</u> : Lit une ligne sous la forme d'un objet.....	1239
<u>8.77.20</u> <u>mysql_fetch_row()</u> : Retourne une ligne sous la forme d'un tableau.....	1239
<u>8.77.21</u> <u>mysql_field_flags()</u> : Lit les options de champ <u>mysql</u>	1239
<u>8.77.22</u> <u>mysql_field_len()</u> : Lit la taille du champ mSQL.....	1240
<u>8.77.23</u> <u>mysql_field_name()</u> : Lit le nom du champ <u>mysql</u>	1240
<u>8.77.24</u> <u>mysql_field_seek()</u> : Fixe l'offset d'un champ mSQL.....	1240
<u>8.77.25</u> <u>mysql_field_table()</u> : Lit le nom de la table du champ <u>mysql</u>	1240
<u>8.77.26</u> <u>mysql_field_type()</u> : Lit le type de champ mSQL.....	1240
<u>8.77.27</u> <u>mysql_fieldflags()</u> : Alias de <u>mysql_field_flags</u>	1240
<u>8.77.28</u> <u>mysql_fieldlen()</u> : Alias de <u>mysql_field_len</u>	1240
<u>8.77.29</u> <u>mysql_fieldname()</u> : Alias de <u>mysql_field_name</u>	1240
<u>8.77.30</u> <u>mysql_fieldtable()</u> : Alias de <u>mysql_field_table</u>	1241
<u>8.77.31</u> <u>mysql_fieldtype()</u> : Alias de <u>mysql_field_type</u>	1241
<u>8.77.32</u> <u>mysql_free_result()</u> : Libère le résultat de la mémoire.....	1241
<u>8.77.33</u> <u>mysql_list_dbs()</u> : Liste les bases de données mSQL sur un serveur.....	1241
<u>8.77.34</u> <u>mysql_list_fields()</u> : Liste les champs d'une table.....	1241
<u>8.77.35</u> <u>mysql_list_tables()</u> : Liste les tables mSQL sur une base de données.....	1241
<u>8.77.36</u> <u>mysql_num_fields()</u> : Retourne le nombre de champs dans un résultat.....	1241
<u>8.77.37</u> <u>mysql_num_rows()</u> : Retourne le nombre de lignes dans un résultat.....	1242
<u>8.77.38</u> <u>mysql_numfields()</u> : Alias de <u>mysql_num_fields</u>	1242
<u>8.77.39</u> <u>mysql_numrows()</u> : Alias de <u>mysql_num_rows</u>	1242
<u>8.77.40</u> <u>mysql_pconnect()</u> : Ouvre une connexion persistante à un serveur mSQL.....	1242
<u>8.77.41</u> <u>mysql_query()</u> : Envoie une requête mSQL.....	1242
<u>8.77.42</u> <u>mysql_regcase()</u> : Alias de <u>sql_regcase</u>	1242
<u>8.77.43</u> <u>mysql_result()</u> : Retourne les données de résultat.....	1242
<u>8.77.44</u> <u>mysql_select_db()</u> : Sélectionne une base de données mSQL.....	1243
<u>8.77.45</u> <u>mysql_tablename()</u> : Alias de <u>mysql_result</u>	1243
<u>8.77.46</u> <u>mysql()</u> : Alias de <u>mysql_db_query</u>	1243
<u>8.78</u> <u>Microsoft SQL Server</u>	1244
<u>8.78.1</u> <u>Introduction</u>	1244
<u>8.78.2</u> <u>Pré-requis</u>	1244
<u>8.78.3</u> <u>Installation</u>	1244
<u>8.78.4</u> <u>Configuration à l'exécution</u>	1244
<u>8.78.5</u> <u>Types de ressources</u>	1245
<u>8.78.6</u> <u>Constantes pré-définies</u>	1245
<u>8.78.8</u> <u>mssql_close()</u> : Ferme une connexion MS SQL Server.....	1246
<u>8.78.9</u> <u>mssql_connect()</u> : Ouvre une connexion à un serveur MS SQL server.....	1246
<u>8.78.10</u> <u>mssql_data_seek()</u> : Déplace le pointeur interne de ligne.....	1247
<u>8.78.11</u> <u>mssql_execute()</u> : Exécute une procédure stockée sur un serveur MS SQL.....	1247
<u>8.78.12</u> <u>mssql_fetch_array()</u> : Lit une ligne de résultat MS SQL dans un tableau.....	1247
<u>8.78.13</u> <u>mssql_fetch_assoc()</u> : Retourne un tableau associatif pour la ligne courant de résultat MS SQL Server.....	1248
<u>8.78.14</u> <u>mssql_fetch_batch()</u> : Retourne le prochain lot de lignes MS SQL Server..	1248
<u>8.78.15</u> <u>mssql_fetch_field()</u> : Lit les informations sur le champ.....	1248

Sommaire

8 Référence des fonctions

<u>8.78.16 mssql_fetch_object()</u> : Retourne une ligne de résultat MS SQL Server sous la forme d'un objet.....	1248
<u>8.78.17 mssql_fetch_row()</u> : Lit une ligne de résultat MS SQL dans un tableau numérique.....	1249
<u>8.78.18 mssql_field_length()</u> : Lit la longueur d'un champ MS SQL Server.....	1249
<u>8.78.19 mssql_field_name()</u> : Lit le nom d'un champ MS SQL Server.....	1249
<u>8.78.20 mssql_field_seek()</u> : Fixe l'offset du pointeur de champ MS SQL Server.....	1250
<u>8.78.21 mssql_field_type()</u> : Lit le nom d'un champ MS SQL Server.....	1250
<u>8.78.22 mssql_free_result()</u> : Libère la mémoire des ressources MS SQL Server.....	1250
<u>8.78.23 mssql_free_statement()</u> : Libère une commande MS SQL Server de la mémoire.....	1250
<u>8.78.24 mssql_get_last_message()</u> : Retourne le dernier message d'erreur du serveur.....	1250
<u>8.78.25 mssql_guid_string()</u> : Convertit le GUID binaire de 16 octets en une chaîne de caractères.....	1251
<u>8.78.26 mssql_init()</u> : Initialise une procédure stockée MS SQL Server locale ou distante.....	1251
<u>8.78.27 mssql_min_error_severity()</u> : Fixe le niveau de sévérité des erreurs MS SQL Server.....	1251
<u>8.78.28 mssql_min_message_severity()</u> : Fixe le niveau de sévérité des messages d'erreurs MS SQL Server.....	1251
<u>8.78.29 mssql_next_result()</u> : Déplace le pointeur interne MS SQL Server au résultat suivant.....	1251
<u>8.78.30 mssql_num_fields()</u> : Retourne le nombre de champs dans un résultat MS SQL Server.....	1252
<u>8.78.31 mssql_num_rows()</u> : Retourne le nombre de lignes dans un résultat MS SQL.....	1252
<u>8.78.32 mssql_pconnect()</u> : Ouvre une connexion persistante à un serveur MS SQL.....	1252
<u>8.78.33 mssql_query()</u> : Envoie une requête SQL au serveur MS SQL.....	1253
<u>8.78.34 mssql_result()</u> : Lit les données d'un résultat.....	1253
<u>8.78.35 mssql_rows_affected()</u> : Retourne le nombre de lignes affectées par une requête MS SQL Server.....	1253
<u>8.78.36 mssql_select_db()</u> : Sélectionne la base de données MS SQL.....	1253
<u>8.79 Fonctions Muscat</u>	1255
<u>8.79.1 Introduction</u>	1255
<u>8.79.2 Installation</u>	1255
<u>8.79.4 muscat_get()</u> : Lit la réponse de l'API muscat.....	1255
<u>8.79.5 muscat_give()</u> : Envoie une chaîne à l'API de base muscat.....	1255
<u>8.79.6 muscat_setup_net()</u> : Crée une nouvelle session muscat.....	1256
<u>8.79.7 muscat_setup()</u> : Crée une nouvelle session.....	1256
<u>8.80 MySQL</u>	1257
<u>8.80.1 Introduction</u>	1257
<u>8.80.2 Pré-requis</u>	1257
<u>8.80.3 Installation</u>	1257
<u>8.80.4 Configuration à l'exécution</u>	1258
<u>8.80.5 Types de ressources</u>	1259
<u>8.80.6 Constantes pré-définies</u>	1259
<u>8.80.7 Notes</u>	1260
<u>8.80.8 Exemples</u>	1260
<u>8.80.10 mysql_change_user()</u> : Change le nom de l'utilisateur de la connexion	

Sommaire

8 Référence des fonctions

<u>active</u>	1262
<u>8.80.11 mysql_client_encoding()</u> : Retourne le nom du jeu de caractères utilisé par le client MySQL.....	1262
<u>8.80.12 mysql_close()</u> : Ferme la connexion MySQL.....	1262
<u>8.80.13 mysql_connect()</u> : Ouvre une connexion à un serveur MySQL.....	1262
<u>8.80.14 mysql_create_db()</u> : Crée une base de données MySQL.....	1263
<u>8.80.15 mysql_data_seek()</u> : Déplace le pointeur interne de résultat MySQL.....	1263
<u>8.80.16 mysql_db_name()</u> : Lit les noms des bases de données.....	1263
<u>8.80.17 mysql_db_query()</u> : Envoie une requête MySQL à un serveur MySQL.....	1263
<u>8.80.18 mysql_drop_db()</u> : Efface une base de données MySQL.....	1263
<u>8.80.19 mysql_errno()</u> : Retourne le numéro d'erreur de la dernière commande MySQL.....	1263
<u>8.80.20 mysql_error()</u> : Retourne le texte associé avec l'erreur générée lors de la dernière requête.....	1264
<u>8.80.21 mysql_escape_string()</u> : Protège une chaîne pour la passer à mysql_query.....	1264
<u>8.80.22 mysql_fetch_array()</u> : Retourne une ligne de résultat MySQL sous la forme d'un tableau associatif, d'un tableau indexé, ou les deux.....	1264
<u>8.80.23 mysql_fetch_assoc()</u> : Lit une ligne de résultat MySQL dans un tableau associatif.....	1264
<u>8.80.24 mysql_fetch_field()</u> : Retourne les données enregistrées dans une colonne MySQL sous forme d'objet.....	1265
<u>8.80.25 mysql_fetch_lengths()</u> : Retourne la taille de chaque colonne d'une ligne de résultat MySQL.....	1265
<u>8.80.26 mysql_fetch_object()</u> : Retourne une ligne de résultat MySQL sous la forme d'un objet.....	1265
<u>8.80.27 mysql_fetch_row()</u> : Retourne une ligne de résultat MySQL sous la forme d'un tableau.....	1265
<u>8.80.28 mysql_field_flags()</u> : Retourne des détails sur une colonne MySQL.....	1265
<u>8.80.29 mysql_field_len()</u> : Retourne la taille d'un champ de résultat MySQL.....	1266
<u>8.80.30 mysql_field_name()</u> : Retourne le nom d'une colonne dans un résultat MySQL.....	1266
<u>8.80.31 mysql_field_seek()</u> : Déplace le pointeur de résultat vers une position donnée.....	1266
<u>8.80.32 mysql_field_table()</u> : Retourne le nom de la table MySQL où se trouve une colonne.....	1266
<u>8.80.33 mysql_field_type()</u> : Retourne le type d'une colonne MySQL spécifique.....	1266
<u>8.80.34 mysql_free_result()</u> : Libère le résultat de la mémoire.....	1267
<u>8.80.35 mysql_get_client_info()</u> : Lit les informations sur le client MySQL.....	1267
<u>8.80.36 mysql_get_host_info()</u> : Lit les informations sur l'hôte MySQL.....	1267
<u>8.80.37 mysql_get_proto_info()</u> : Lit les informations sur le protocole MySQL.....	1267
<u>8.80.38 mysql_get_server_info()</u> : Lit les informations sur le serveur MySQL.....	1267
<u>8.80.39 mysql_info()</u> : Lit des informations à propos de la dernière requête MySQL.....	1267
<u>8.80.40 mysql_insert_id()</u> : Retourne l'identifiant généré par la dernière requête INSERT MySQL.....	1268
<u>8.80.41 mysql_list_dbs()</u> : Liste les bases de données disponibles sur le serveur MySQL.....	1268
<u>8.80.42 mysql_list_fields()</u> : Liste les champs d'une table MySQL.....	1268
<u>8.80.43 mysql_list_processes()</u> : Liste les processus MySQL.....	1268
<u>8.80.44 mysql_list_tables()</u> : Liste les tables d'une base de données MySQL.....	1268

Sommaire

8 Référence des fonctions

<u>8.80.45 mysql_num_fields() : Retourne le nombre de champs d'un résultat MySQL</u>	1268
<u>8.80.46 mysql_num_rows() : Retourne le nombre de lignes d'un résultat MySQL</u>	1269
<u>8.80.47 mysql_pconnect() : Ouvre une connexion persistante à un serveur MySQL</u>	1269
<u>8.80.48 mysql_ping() : Pingue le serveur MySQL, et s'y reconnecte au besoin</u>	1269
<u>8.80.49 mysql_query() : Envoi une requête à un serveur MySQL</u>	1269
<u>8.80.50 mysql_real_escape_string() : Protège les caractères spéciaux d'une commande SQL</u>	1270
<u>8.80.51 mysql_result() : Retourne un champ d'un résultat MySQL</u>	1270
<u>8.80.52 mysql_select_db() : Sélectionne une base de données MySQL</u>	1270
<u>8.80.53 mysql_stat() : Retourne le statut courant du serveur MySQL</u>	1270
<u>8.80.54 mysql_tablename() : Lit le nom de la table qui contient un champ</u>	1270
<u>8.80.55 mysql_thread_id() : Retourne l'identifiant du thread MySQL courant</u>	1271
<u>8.80.56 mysql_unbuffered_query() : Exécute une requête SQL sans mobiliser les résultats MySQL</u>	1271
<u>8.81 Extension MySQL améliorée</u>	1272
<u>8.81.1 Introduction</u>	1272
<u>8.81.2 Pré-requis</u>	1272
<u>8.81.3 Installation</u>	1272
<u>8.81.4 Configuration à l'exécution</u>	1272
<u>8.81.5 Classes pré-définies</u>	1273
<u>8.81.6 Constantes pré-définies</u>	1275
<u>8.81.8 mysqli_autocommit() : mysqli->autocommit()</u>	1279
<u>8.81.9 mysqli_bind_param() : Alias de mysqli_stmt_bind_param</u>	1280
<u>8.81.10 mysqli_bind_result() : Alias de mysqli_stmt_bind_result</u>	1280
<u>8.81.11 mysqli_change_user() : mysqli->change_user()</u>	1280
<u>8.81.12 mysqli_character_set_name() : mysqli->character_set_name()</u>	1280
<u>8.81.13 mysqli_client_encoding() : Alias de mysqli_character_set_name</u>	1281
<u>8.81.14 mysqli_close() : mysqli->close()</u>	1281
<u>8.81.15 mysqli_commit() : mysqli->commit()</u>	1281
<u>8.81.16 mysqli_connect_errno() : Retourne le code d'erreur de la connexion MySQL</u>	1281
<u>8.81.17 mysqli_connect_error() : Retourne le message d'erreur de connexion MySQL</u>	1282
<u>8.81.18 mysqli_connect() : mysqli()</u>	1282
<u>8.81.19 mysqli_data_seek() : result->data_seek()</u>	1282
<u>8.81.20 mysqli_debug() : Effectue des actions de débogage</u>	1283
<u>8.81.21</u>	
<u>mysqli_disable_reads_from_master() : mysqli->disable_reads_from_master()</u>	1283
<u>8.81.22 mysqli_disable_rpl_parse() : Désactive le parseur RPL</u>	1283
<u>8.81.23 mysqli_dump_debug_info() : mysqli->dump_debug_info()</u>	1283
<u>8.81.24 mysqli_embedded_connect() : Ouvre une connexion avec un serveur MySQL intégré</u>	1284
<u>8.81.25 mysqli_enable_reads_from_master() : Active la lecture depuis le maître</u>	1284
<u>8.81.26 mysqli_enable_rpl_parse() : Active le parseur RPL</u>	1284
<u>8.81.27 mysqli_errno() : mysqli->errno()</u>	1284
<u>8.81.28 mysqli_error() : Retourne une chaîne décrivant la dernière erreur</u>	1284
<u>8.81.29 mysqli_escape_string() : Alias de mysqli_real_escape_string</u>	1285
<u>8.81.30 mysqli_execute() : Alias de mysqli_stmt_execute</u>	1285
<u>8.81.31 mysqli_fetch_array() : result->fetch_array()</u>	1285

Sommaire

8 Référence des fonctions

<u>8.81.32</u> <u>mysqli_fetch_assoc()</u> : <u>mysqli->fetch_assoc()</u>	1286
<u>8.81.33</u> <u>mysqli_fetch_field_direct()</u> : <u>result->fetch_field_direct()</u>	1286
<u>8.81.34</u> <u>mysqli_fetch_field()</u> : <u>result->fetch_field()</u>	1286
<u>8.81.35</u> <u>mysqli_fetch_fields()</u> : <u>result->fetch_fields()</u>	1287
<u>8.81.36</u> <u>mysqli_fetch_lengths()</u> : <u>result->lengths()</u>	1287
<u>8.81.37</u> <u>mysqli_fetch_object()</u> : <u>result->fetch_object()</u>	1287
<u>8.81.38</u> <u>mysqli_fetch_row()</u> : <u>result->fetch_row()</u>	1288
<u>8.81.39</u> <u>mysqli_fetch()</u> : Alias de <u>mysqli_stmt_fetch</u>	1288
<u>8.81.40</u> <u>mysqli_field_count()</u> : <u>mysqli->field_count()</u>	1288
<u>8.81.41</u> <u>mysqli_field_seek()</u> : <u>result->field_seek()</u>	1288
<u>8.81.42</u> <u>mysqli_field_tell()</u> : <u>result->current_field()</u>	1289
<u>8.81.43</u> <u>mysqli_free_result()</u> : <u>result->free()</u>	1289
<u>8.81.44</u> <u>mysqli_get_client_info()</u> : Retourne une chaîne contenant la version du client MySQL.....	1289
<u>8.81.45</u> <u>mysqli_get_client_version()</u> : Lit les informations du client MySQL.....	1289
<u>8.81.46</u> <u>mysqli_get_host_info()</u> : <u>mysqli->get_host_info()</u>	1290
<u>8.81.47</u> <u>mysqli_get_metadata()</u> : Alias de <u>mysqli_stmt_result_metadata</u>	1290
<u>8.81.48</u> <u>mysqli_get_proto_info()</u> : <u>mysqli->protocol_version()</u>	1290
<u>8.81.49</u> <u>mysqli_get_server_info()</u> : <u>mysqli->server_info()</u>	1290
<u>8.81.50</u> <u>mysqli_get_server_version()</u> : Retourne un entier représentant la version du serveur MySQL.....	1291
<u>8.81.51</u> <u>mysqli_info()</u> : <u>mysqli->info()</u>	1291
<u>8.81.52</u> <u>mysqli_init()</u> : Initialise MySQLi et retourne une ressource à utiliser avec <u>mysqli_real_connect()</u>	1291
<u>8.81.53</u> <u>mysqli_insert_id()</u> : <u>mysqli->insert_id()</u>	1292
<u>8.81.54</u> <u>mysqli_kill()</u> : <u>mysqli->kill()</u>	1292
<u>8.81.55</u> <u>mysqli_master_query()</u> : Force l'exécution d'une requête sur le maître dans une configuration maître/esclave (<u>master/slave</u>).....	1292
<u>8.81.56</u> <u>mysqli_more_results()</u> : <u>mysqli->more_results()</u>	1292
<u>8.81.57</u> <u>mysqli_multi_query()</u> : <u>mysqli->multi_query()</u>	1293
<u>8.81.58</u> <u>mysqli_next_result()</u> : <u>mysqli->next_result()</u>	1293
<u>8.81.59</u> <u>mysqli_num_fields()</u> : <u>result->field_count()</u>	1293
<u>8.81.60</u> <u>mysqli_num_rows()</u> : Retourne le nombre de lignes dans un résultat.....	1293
<u>8.81.61</u> <u>mysqli_options()</u> : <u>mysqli->options()</u>	1294
<u>8.81.62</u> <u>mysqli_param_count()</u> : Alias de <u>mysqli_stmt_param_count</u>	1294
<u>8.81.63</u> <u>mysqli_ping()</u> : <u>mysqli->ping()</u>	1294
<u>8.81.64</u> <u>mysqli_prepare()</u> : <u>mysqli->prepare()</u>	1295
<u>8.81.65</u> <u>mysqli_query()</u> : <u>mysqli->query()</u>	1295
<u>8.81.66</u> <u>mysqli_real_connect()</u> : <u>mysqli->real_connect()</u>	1296
<u>8.81.67</u> <u>mysqli_real_escape_string()</u> : <u>mysqli->real_escape_string()</u>	1297
<u>8.81.68</u> <u>mysqli_real_query()</u> : <u>mysqli->real_query()</u>	1297
<u>8.81.69</u> <u>mysqli_report()</u> : Active ou désactive les fonctions de rapport interne.....	1297
<u>8.81.70</u> <u>mysqli_rollback()</u> : <u>mysqli->rollback()</u>	1298
<u>8.81.71</u> <u>mysqli_rpl_parse_enabled()</u> : Vérifie si le parseur RPL est activé.....	1298
<u>8.81.72</u> <u>mysqli_rpl_probe()</u> : Sonde le RPL.....	1298
<u>8.81.73</u> <u>mysqli_rpl_query_type()</u> : <u>mysqli->rpl_query_type()</u>	1298
<u>8.81.74</u> <u>mysqli_select_db()</u> : <u>mysqli->select_db()</u>	1298
<u>8.81.75</u> <u>mysqli_send_long_data()</u> : Alias de <u>mysqli_stmt_send_long_data</u>	1299
<u>8.81.76</u> <u>mysqli_send_query()</u> : <u>mysqli->send_query()</u>	1299
<u>8.81.77</u> <u>mysqli_server_end()</u> : Eteint le serveur embarqué.....	1299
<u>8.81.78</u> <u>mysqli_server_init()</u> : Initialise le serveur MySQL intégré.....	1299

Sommaire

8 Référence des fonctions

<u>8.81.79</u>	<u>mysql_set_charset() : mysql->set_charset()</u>	1299
<u>8.81.80</u>	<u>mysql_set_opt() : Alias de mysql_options</u>	1300
<u>8.81.81</u>	<u>mysql_sqlstate() : mysql->sqlstate()</u>	1300
<u>8.81.82</u>	<u>mysql_ssl_set() : mysql->ssl_set()</u>	1300
<u>8.81.83</u>	<u>mysql_stat() : mysql->stat()</u>	1300
<u>8.81.84</u>	<u>mysql_stmt_affected_rows() : mysql_stmt->affected_rows()</u>	1301
<u>8.81.85</u>	<u>mysql_stmt_bind_param() : stmt->bind_param()</u>	1301
<u>8.81.86</u>	<u>mysql_stmt_bind_result() : stmt->bind_result()</u>	1302
<u>8.81.87</u>	<u>mysql_stmt_close() : mysql_stmt->close()</u>	1302
<u>8.81.88</u>	<u>mysql_stmt_data_seek() : stmt->data_seek()</u>	1302
<u>8.81.89</u>	<u>mysql_stmt_errno() : mysql_stmt->errno()</u>	1303
<u>8.81.90</u>	<u>mysql_stmt_error() : mysql_stmt->error()</u>	1303
<u>8.81.91</u>	<u>mysql_stmt_execute() : stmt->execute()</u>	1303
<u>8.81.92</u>	<u>mysql_stmt_fetch() : stmt->fetch()</u>	1304
<u>8.81.93</u>	<u>mysql_stmt_free_result() : stmt->free_result()</u>	1304
<u>8.81.94</u>	<u>mysql_stmt_init() : mysql->stmt_init()</u>	1304
<u>8.81.95</u>	<u>mysql_stmt_num_rows() : stmt->num_rows()</u>	1305
<u>8.81.96</u>	<u>mysql_stmt_param_count() : stmt->param_count()</u>	1305
<u>8.81.97</u>	<u>mysql_stmt_prepare() : stmt->prepare()</u>	1305
<u>8.81.98</u>	<u>mysql_stmt_reset() : stmt->reset()</u>	1306
<u>8.81.99</u>	<u>mysql_stmt_result_metadata() : Retourne les méta-données de préparation de requête MySQL</u>	1306
<u>8.81.100</u>	<u>mysql_stmt_send_long_data() : stmt->send_long_data()</u>	1307
<u>8.81.101</u>	<u>mysql_stmt_sqlstate() : Retourne le code SQLSTATE de la dernière opération MySQL</u>	1307
<u>8.81.102</u>	<u>mysql_stmt_store_result() : mysql_stmt->store_result()</u>	1307
<u>8.81.103</u>	<u>mysql_store_result() : mysql->store_result()</u>	1308
<u>8.81.104</u>	<u>mysql_thread_id() : mysql->thread_id()</u>	1308
<u>8.81.105</u>	<u>mysql_thread_safe() : Indique si le thread safety est donné ou non</u>	1309
<u>8.81.106</u>	<u>mysql_use_result() : mysql->use_result()</u>	1309
<u>8.81.107</u>	<u>mysql_warning_count() : mysql->warning_count()</u>	1309
<u>8.82</u>	<u>Fonctions de contrôle d'écran de terminal</u>	1311
<u>8.82.1</u>	<u>Introduction</u>	1311
<u>8.82.2</u>	<u>Pré-requis</u>	1311
<u>8.82.3</u>	<u>Installation</u>	1311
<u>8.82.4</u>	<u>Configuration à l'exécution</u>	1311
<u>8.82.5</u>	<u>Types de ressources</u>	1312
<u>8.82.6</u>	<u>Constantes pré-définies</u>	1312
<u>8.82.8</u>	<u>ncurses_addchnstr() : Ajoute une chaîne de taille donnée à la position courante</u>	1318
<u>8.82.9</u>	<u>ncurses_addchstr() : Ajoute une chaîne à la position courante</u>	1318
<u>8.82.10</u>	<u>ncurses_addnstr() : Ajoute une chaîne à la position courante</u>	1318
<u>8.82.11</u>	<u>ncurses_addstr() : Affiche du texte à la position courante</u>	1318
<u>8.82.12</u>	<u>ncurses_assume_default_colors() : Définit la couleur 0</u>	1319
<u>8.82.13</u>	<u>ncurses_attroff() : Désactive les attributs donnés</u>	1319
<u>8.82.14</u>	<u>ncurses_attron() : Active les attributs suivants</u>	1319
<u>8.82.15</u>	<u>ncurses_attrset() : Modifie les attributs donnés</u>	1319
<u>8.82.16</u>	<u>ncurses_baudrate() : Retourne le baudrate du terminal</u>	1320
<u>8.82.17</u>	<u>ncurses_beep() : Fait beeper le terminal</u>	1320
<u>8.82.18</u>	<u>ncurses_bkgd() : Définit les propriétés de fond d'écran pour le terminal</u>	1320
<u>8.82.19</u>	<u>ncurses_bkgdset() : Définit le fond d'écran</u>	1320

Sommaire

8 Référence des fonctions

<u>8.82.20 ncurses</u> <u>border()</u> : Dessine un bord autour de l'écran avec les caractères données.....	1321
<u>8.82.21 ncurses</u> <u>bottom_panel()</u> : Place un conteneur visible en bas de la pile.....	1321
<u>8.82.22 ncurses</u> <u>can_change_color()</u> : Vérifie si le terminal peut changer de couleurs.....	1321
<u>8.82.23 ncurses</u> <u>cbreak()</u> : Change la bufferisation d'entrée.....	1321
<u>8.82.24 ncurses</u> <u>clear()</u> : Efface l'écran.....	1322
<u>8.82.25 ncurses</u> <u>clrtoebot()</u> : Efface l'écran depuis la position courante jusqu'au bas de l'écran.....	1322
<u>8.82.26 ncurses</u> <u>clrtoeol()</u> : Efface l'écran depuis la position courante jusqu'à la fin de la ligne.....	1322
<u>8.82.27 ncurses</u> <u>color_content()</u> : Lit la valeur RGB d'une couleur.....	1323
<u>8.82.28 ncurses</u> <u>color_set()</u> : Modifie la couleur de fond et de devant.....	1323
<u>8.82.29 ncurses</u> <u>cur_set()</u> : Modifie l'état du curseur.....	1323
<u>8.82.30 ncurses</u> <u>def_prog_mode()</u> : Sauve le mode du terminal.....	1323
<u>8.82.31 ncurses</u> <u>def_shell_mode()</u> : Sauve le mode de terminal (shell).....	1323
<u>8.82.32 ncurses</u> <u>define_key()</u> : Définit un code de clé (keycode).....	1324
<u>8.82.33 ncurses</u> <u>del_panel()</u> : Retire un conteneur de la pile et l'efface (mais pas la fenêtre associée).....	1324
<u>8.82.34 ncurses</u> <u>delay_output()</u> : Retarde l'affichage sur les terminaux utilisant des caractères de remplissage.....	1324
<u>8.82.35 ncurses</u> <u>delch()</u> : Efface le caractère courant et décale le reste de la ligne vers la gauche.....	1324
<u>8.82.36 ncurses</u> <u>deleteln()</u> : Efface la ligne courante, et monte l'écran d'une ligne.....	1325
<u>8.82.37 ncurses</u> <u>delwin()</u> : Efface une fenêtre ncurses.....	1325
<u>8.82.38 ncurses</u> <u>doupdate()</u> : Affiche tout ce qui est prêt, et rafraîchit l'écran.....	1325
<u>8.82.39 ncurses</u> <u>echo()</u> : Active l'écho d'entrée clavier.....	1326
<u>8.82.40 ncurses</u> <u>echochar()</u> : Affiche un caractère et rafraîchit l'écran.....	1326
<u>8.82.41 ncurses</u> <u>end()</u> : Cesse l'utilisation de ncurses et efface l'écran.....	1326
<u>8.82.42 ncurses</u> <u>erase()</u> : Efface l'écran du terminal.....	1326
<u>8.82.43 ncurses</u> <u>erasechar()</u> : Lit le caractère qui se fait effacer.....	1327
<u>8.82.44 ncurses</u> <u>filter()</u> : Définit LINES pour inisr() et newterm() à 1.....	1327
<u>8.82.45 ncurses</u> <u>flash()</u> : Fait flasher le terminal (visual bell).....	1327
<u>8.82.46 ncurses</u> <u>flushinp()</u> : Vide le buffer d'entrée du clavier.....	1327
<u>8.82.47 ncurses</u> <u>getch()</u> : Lit un caractère sur le clavier.....	1328
<u>8.82.48 ncurses</u> <u>getmaxyx()</u> : Retourne la taille d'une fenêtre.....	1328
<u>8.82.49 ncurses</u> <u>getmouse()</u> : Lit les événements souris.....	1328
<u>8.82.50 ncurses</u> <u>getyx()</u> : Retourne la position courante du curseur pour une fenêtre.....	1329
<u>8.82.51 ncurses</u> <u>halfdelay()</u> : Met le terminal en mode semi-retardé.....	1329
<u>8.82.52 ncurses</u> <u>has_colors()</u> : Vérifie que le terminal supporte les couleurs.....	1329
<u>8.82.53 ncurses</u> <u>has_ic()</u> : Vérifie les capacités d'insertion et d'effacement.....	1329
<u>8.82.54 ncurses</u> <u>has_il()</u> : Vérifie les capacités d'insertion et d'effacement.....	1330
<u>8.82.55 ncurses</u> <u>has_key()</u> : Vérifie la présence des touches de fonctions sur le clavier.....	1330
<u>8.82.56 ncurses</u> <u>hide_panel()</u> : Retire un conteneur de la pile, pour le rendre invisible.....	1330
<u>8.82.57 ncurses</u> <u>hline()</u> : Dessine une ligne horizontale à la position courante, en utilisant un caractère et une taille maximale.....	1330
<u>8.82.58 ncurses</u> <u>inch()</u> : Lit le caractère et ses attributs à la position courante.....	1331
<u>8.82.59 ncurses</u> <u>init_color()</u> : Configure une nouvelle valeur RGB pour une	

Sommaire

8 Référence des fonctions

<u>couleur</u>	1331
<u>8.82.60 ncurses_init_pair()</u> : Alloue une paire de couleur.....	1331
<u>8.82.61 ncurses_init()</u> : Initialise ncurses.....	1331
<u>8.82.62 ncurses_insch()</u> : Insère un caractère et décale le reste de la ligne vers la droite, y compris le caractère courant.....	1332
<u>8.82.63 ncurses_insdelln()</u> : Insère des lignes devant la ligne courante en scrollant vers le bas (des nombres négatifs donneront un scroll vers le haut).....	1332
<u>8.82.64 ncurses_insertln()</u> : Insère une ligne et décale le reste de l'écran vers le bas.....	1332
<u>8.82.65 ncurses_insstr()</u> : Insère une chaîne à la position courante, et décale le reste de la chaîne à droite.....	1333
<u>8.82.66 ncurses_instr()</u> : Lit une chaîne depuis le terminal.....	1333
<u>8.82.67 ncurses_isendwin()</u> : Ncurses est en mode endwin, des affichages normaux sont possibles.....	1333
<u>8.82.68 ncurses_keyok()</u> : Active ou désactive un code de clé (keycode).....	1333
<u>8.82.69 ncurses_keypad()</u> : Active ou désactive le keypad.....	1334
<u>8.82.70 ncurses_killchar()</u> : Retourne la ligne du caractère actuellement supprimé.....	1334
<u>8.82.71 ncurses_longname()</u> : Retourne la description du terminal.....	1334
<u>8.82.72 ncurses_meta()</u> : Active/désactive les informations de méta clé 8-bits.....	1334
<u>8.82.73 ncurses_mouse_trafo()</u> : Transforme les coordonnées.....	1334
<u>8.82.74 ncurses_mouseinterval()</u> : Configure les délais entre les clics de souris.....	1335
<u>8.82.75 ncurses_mousemask()</u> : Configure les événements de souris à surveiller....	1335
<u>8.82.76 ncurses_move_panel()</u> : Déplace un conteneur pour que son coin supérieur gauche soit aux coordonnées [startx, starty].....	1336
<u>8.82.77 ncurses_move()</u> : Déplace la position d'affichage.....	1336
<u>8.82.78 ncurses_mvaddch()</u> : Déplace la position courante et ajoute un caractère...	1336
<u>8.82.79 ncurses_mvaddchnstr()</u> : Déplace la position et ajoute la chaîne attribuée avec la taille donnée.....	1337
<u>8.82.80 ncurses_mvaddchstr()</u> : Déplace la position et ajoute une chaîne attribuée.....	1337
<u>8.82.81 ncurses_mvaddnstr()</u> : Déplace la position et ajoute une chaîne de taille donnée.....	1337
<u>8.82.82 ncurses_mvaddstr()</u> : Déplace la position et ajoute une chaîne.....	1337
<u>8.82.83 ncurses_mvcur()</u> : Déplace immédiatement le curseur.....	1338
<u>8.82.84 ncurses_mvdelch()</u> : Déplace la position et efface le caractère, puis décale le reste de la ligne à gauche.....	1338
<u>8.82.85 ncurses_mvgetch()</u> : Déplace la position et lit le caractère à la nouvelle position.....	1338
<u>8.82.86 ncurses_mvhline()</u> : Choisi une nouvelle position et dessine une ligne horizontale avec le caractère donné, et de taille maximale donnée.....	1338
<u>8.82.87 ncurses_mvinch()</u> : Déplace la position et lit le caractère à la nouvelle position.....	1339
<u>8.82.88 ncurses_mvvline()</u> : Modifie la position et dessine une ligne verticale avec le caractère donné, de taille maximale donnée.....	1339
<u>8.82.89 ncurses_mvwaddstr()</u> : Ajoute une chaîne à une nouvelle position dans la fenêtre.....	1339
<u>8.82.90 ncurses_napms()</u> : Fait une pause.....	1339
<u>8.82.91 ncurses_new_panel()</u> : Crée un nouveau conteneur et l'associe à une fenêtre.....	1340
<u>8.82.92 ncurses_newpad()</u> : Crée un nouveau pad (fenêtre).....	1340
<u>8.82.93 ncurses_newwin()</u> : Crée une nouvelle fenêtre.....	1340

Sommaire

8 Référence des fonctions

<u>8.82.94 ncurses_nl()</u> : Convertit les nouvelles lignes et retours chariot en sauts de ligne.....	1340
<u>8.82.95 ncurses_nocbreak()</u> : Passe le terminal en mode cooked.....	1341
<u>8.82.96 ncurses_noecho()</u> : Désactive l'écho clavier.....	1341
<u>8.82.97 ncurses_nonl()</u> : Ne convertit pas les nouvelles lignes et retours chariot en sauts de ligne.....	1341
<u>8.82.98 ncurses_noqiflush()</u> : Ne vide pas les buffers sur réception des caractères de signaux.....	1341
<u>8.82.99 ncurses_noraw()</u> : Déconfigure le mode brut du terminal.....	1342
<u>8.82.100 ncurses_pair_content()</u> : Retourne la valeur RGB d'une couleur.....	1342
<u>8.82.101 ncurses_panel_above()</u> : Retourne le conteneur au-dessus du conteneur.....	1342
<u>8.82.102 ncurses_panel_below()</u> : Retourne le conteneur sous le conteneur.....	1342
<u>8.82.103 ncurses_panel_window()</u> : Retourne la fenêtre associée avec un conteneur.....	1343
<u>8.82.104 ncurses_pnoutrefresh()</u> : Copie une région depuis un pad dans un écran virtuel.....	1343
<u>8.82.105 ncurses_prefresh()</u> : Copie une région depuis un pad dans l'écran virtuel.....	1343
<u>8.82.106 ncurses_putp()</u> : Applique l'information d'espacement à la chaîne et l'affiche.....	1343
<u>8.82.107 ncurses_qiflush()</u> : Vide les buffers après détection d'un caractère de signal.....	1343
<u>8.82.108 ncurses_raw()</u> : Passe le terminal en mode brut (raw).....	1344
<u>8.82.109 ncurses_refresh()</u> : Rafraîchit l'écran.....	1344
<u>8.82.110 ncurses_replace_panel()</u> : Remplace la fenêtre associée à un conteneur.....	1344
<u>8.82.111 ncurses_reset_prog_mode()</u> : Restaure le mode prog sauvé par def_prog_mode.....	1344
<u>8.82.112 ncurses_reset_shell_mode()</u> : Restaure le mode shell, mode sauvé par def_shell_mode.....	1345
<u>8.82.113 ncurses_resetty()</u> : Rétablit le terminal sauvé.....	1345
<u>8.82.114 ncurses_savetty()</u> : Sauve l'état du terminal.....	1345
<u>8.82.115 ncurses_scr_dump()</u> : Sauve un écran dans un fichier.....	1345
<u>8.82.116 ncurses_scr_init()</u> : Initialise un écran depuis un fichier de sauvegarde.....	1346
<u>8.82.117 ncurses_scr_restore()</u> : Rétablit un écran depuis un fichier de sauvegarde.....	1346
<u>8.82.118 ncurses_scr_set()</u> : Configure un écran depuis un fichier de sauvegarde.....	1346
<u>8.82.119 ncurses_scrll()</u> : Scrolle le contenu de la fenêtre vers le haut ou le bas, sans changer la position courante.....	1346
<u>8.82.120 ncurses_show_panel()</u> : Place un conteneur invisible en haut de la pile, pour le rendre visible.....	1347
<u>8.82.121 ncurses_slk_attr()</u> : Retourne l'attribut de la clé soft label courante.....	1347
<u>8.82.122 ncurses_slk_attroff()</u> : Désactive l'attribut donné pour les étiquettes des function-key (fonctions clés).....	1347
<u>8.82.123 ncurses_slk_attron()</u> : Active l'attribut donné pour les étiquettes des function-key (fonctions clés).....	1347
<u>8.82.124 ncurses_slk_attrset()</u> : Définit les attributs donnés à une étiquette function-key.....	1348
<u>8.82.125 ncurses_slk_clear()</u> : Efface les soft labels de l'écran.....	1348
<u>8.82.126 ncurses_slk_color()</u> : Configure la couleur des clés soft label.....	1348

Sommaire

8 Référence des fonctions

<u>8.82.127 ncurses_slk_init()</u> : Initialise les fonctions de clés soft label.....	1348
<u>8.82.128 ncurses_slk_noutrefresh()</u> : Copie toutes les clés soft label à l'écran virtuel.....	1349
<u>8.82.129 ncurses_slk_refresh()</u> : Copie les clés soft label à l'écran.....	1349
<u>8.82.130 ncurses_slk_restore()</u> : Rétablit les clés soft label.....	1349
<u>8.82.131 ncurses_slk_set()</u> : Modifie les étiquettes de clé de fonction (fonction key labels).....	1349
<u>8.82.132 ncurses_slk_touch()</u> : Force l'affichage lorsque ncurses_slk_noutrefresh est utilisé.....	1349
<u>8.82.133 ncurses_standend()</u> : Cesse l'utilisation de l'attribut 'standout'.....	1350
<u>8.82.134 ncurses_standout()</u> : Commence l'utilisation de l'attribut 'standout'.....	1350
<u>8.82.135 ncurses_start_color()</u> : Active les couleurs.....	1350
<u>8.82.136 ncurses_termattrs()</u> : Retourne toutes les options supportées par le terminal, combinées ensemble par l'opérateur OR.....	1350
<u>8.82.137 ncurses_termname()</u> : Retourne le nom court du terminal.....	1351
<u>8.82.138 ncurses_timeout()</u> : Configure le temps d'expiration des séquences clavier spéciales.....	1351
<u>8.82.139 ncurses_top_panel()</u> : Place un conteneur visible sur le haut de la pile.....	1351
<u>8.82.140 ncurses_ttypeahead()</u> : Spécifie un autre descripteur de fichier pour la vérification à la volée des données saisies.....	1351
<u>8.82.141 ncurses_ungetch()</u> : Remplace un caractère dans la queue d'entrée.....	1352
<u>8.82.142 ncurses_ungetmouse()</u> : Ajoute un événement souris dans la queue.....	1352
<u>8.82.143 ncurses_update_panels()</u> : Rafraîchit l'écran virtuel pour prendre en compte les relations entre les conteneurs de la pile.....	1352
<u>8.82.144 ncurses_use_default_colors()</u> : Assigne la couleur de terminal pour l'index -1.....	1353
<u>8.82.145 ncurses_use_env()</u> : Contrôle l'utilisation des informations d'environnement concernant la taille du terminal.....	1353
<u>8.82.146 ncurses_use_extended_names()</u> : Contrôle l'utilisation des noms étendus dans les descriptions terminfo.....	1353
<u>8.82.147 ncurses_vidattr()</u> : Affiche la chaîne sur un terminal dans le mode d'attribut vidéo.....	1353
<u>8.82.148 ncurses_vline()</u> : Dessine une ligne verticale à la position courante en utilisant un caractère donné, pour une taille donnée.....	1354
<u>8.82.149 ncurses_waddch()</u> : Ajoute un caractère à la position courante, dans une fenêtre, et avance le curseur.....	1354
<u>8.82.150 ncurses_waddstr()</u> : Affiche le texte à la position courante dans la fenêtre.....	1354
<u>8.82.151 ncurses_wattroff()</u> : Désactive les attributs d'une fenêtre.....	1354
<u>8.82.152 ncurses_wattron()</u> : Efface les attributs de la fenêtre.....	1354
<u>8.82.153 ncurses_wattrset()</u> : Modifie les attributs d'une fenêtre.....	1355
<u>8.82.154 ncurses_wborder()</u> : Dessine le bord d'une fenêtre avec un caractère qualifié.....	1355
<u>8.82.155 ncurses_wclear()</u> : Efface la fenêtre.....	1355
<u>8.82.156 ncurses_wcolor_set()</u> : Modifie les paires de couleurs de la fenêtre.....	1355
<u>8.82.157 ncurses_werase()</u> : Efface le contenu de la fenêtre.....	1355
<u>8.82.158 ncurses_wgetch()</u> : Lit un caractère depuis le clavier (fenêtre).....	1355
<u>8.82.159 ncurses_whline()</u> : Dessine une ligne horizontale à partir de la position courante, avec un caractère qualifié, et d'une taille maximale.....	1356
<u>8.82.160 ncurses_wmouse_trafo()</u> : Transforme les coordonnées d'une fenêtre.....	1356
<u>8.82.161 ncurses_wmove()</u> : Déplace la position d'affichage de la fenêtre.....	1356

Sommaire

8 Référence des fonctions

<u>8.82.162 ncurses_wnoutrefresh()</u> : Copie la fenêtre dans l'écran virtuel.....	1356
<u>8.82.163 ncurses_wrefresh()</u> : Rafraîchit la fenêtre sur l'écran du terminal.....	1356
<u>8.82.164 ncurses_wstandend()</u> : Termine le mode standout de la fenêtre.....	1357
<u>8.82.165 ncurses_wstandout()</u> : Passe une fenêtre en mode standout.....	1357
<u>8.82.166 ncurses_wvline()</u> : Dessine une ligne verticale dans une fenêtre, à la position courante, en utilisant un caractère qualifié, avec une taille maximale....	1357
8.83 Réseau.....	1358
<u>8.83.1 Introduction.....</u>	<u>1358</u>
<u>8.83.2 Pré-requis.....</u>	<u>1358</u>
<u>8.83.3 Installation.....</u>	<u>1358</u>
<u>8.83.4 Configuration à l'exécution.....</u>	<u>1358</u>
<u>8.83.5 Types de ressources.....</u>	<u>1358</u>
<u>8.83.6 Constantes pré-définies.....</u>	<u>1358</u>
<u>8.83.8 closelog()</u> : Ferme la connexion à l'historique système.....	1360
<u>8.83.9 debugger_off()</u> : Désactive le débogueur interne de PHP (PHP 3).....	1360
<u>8.83.10 debugger_on()</u> : Active le débogueur interne de PHP (PHP 3).....	1360
<u>8.83.11 define_syslog_variables()</u> : Initialise toutes les constantes liées au syslog..	1361
<u>8.83.12 dns_check_record()</u> : Synonyme de checkdnrr.....	1361
<u>8.83.13 dns_get_mx()</u> : Synonyme de getmxrr.....	1361
<u>8.83.14 dns_get_record()</u> : Lit les données DNS associées à un hôte.....	1361
<u>8.83.15 fsocketopen()</u> : Ouvre une socket de connexion Internet ou Unix.....	1366
<u>8.83.16 gethostbyaddr()</u> : Retourne le nom d'hôte correspondant à une IP.....	1367
<u>8.83.17 gethostbyname()</u> : Retourne l'adresse IP correspondant à un hôte.....	1367
<u>8.83.18 gethostbyname1()</u> : Retourne la liste d'IP correspondante à un hôte.....	1368
<u>8.83.19 getmxrr()</u> : Retourne les enregistrements MX d'un hôte.....	1368
<u>8.83.20 getprotobyname()</u> : Retourne le numéro de protocole associé à un nom de protocole.....	1369
<u>8.83.21 getprotobyname1()</u> : Retourne le nom de protocole associé à un numéro de protocole.....	1369
<u>8.83.22 getservbyname()</u> : Retourne le numéro de port associé à un service Internet et un protocole.....	1369
<u>8.83.23 getservbyport()</u> : Retourne le service Internet qui correspond au port et protocole.....	1370
<u>8.83.24 inet_ntop()</u> : Convertit un paquet d'adresses internet en une représentation humainement lisible.....	1370
<u>8.83.25 inet_pton()</u> : Convertit une adresse IP lisible en sa représentation in_addr.....	1370
<u>8.83.26 ip2long()</u> : Convertit une chaîne contenant une adresse (IPv4) IP numérique en adresse littérale.....	1371
<u>8.83.27 long2ip()</u> : Convertit une adresse IP (IPv4) en adresse IP numérique.....	1372
<u>8.83.28 openlog()</u> : Ouvre la connexion à l'historique système.....	1372
<u>8.83.29 psocketopen()</u> : Ouvre une socket de connexion Internet ou Unix persistante.....	1373
<u>8.83.30 socket_get_status()</u> : Alias de stream_get_meta_data.....	1373
<u>8.83.31 socket_set_blocking()</u> : Alias de stream_set_blocking.....	1373
<u>8.83.32 socket_set_timeout()</u> : Alias de stream_set_timeout.....	1373
<u>8.83.33 syslog()</u> : Génère un message dans l'historique système.....	1373
8.84 NIS.....	1375
<u>8.84.1 Introduction.....</u>	<u>1375</u>
<u>8.84.2 Pré-requis.....</u>	<u>1375</u>
<u>8.84.3 Installation.....</u>	<u>1375</u>

Sommaire

8 Référence des fonctions

<u>8.84.4 Configuration à l'exécution</u>	1375
<u>8.84.5 Constantes pré-définies</u>	1375
<u>8.84.7 yp_cat()</u> : Retourne une tableau contenant la carte entière.....	1376
<u>8.84.8 yp_err_string()</u> : Retourne la chaîne correspondant à l'erreur donnée.....	1376
<u>8.84.9 yp_errno()</u> : Retourne le code d'erreur de la dernière opération NIS.....	1376
<u>8.84.10 yp_first()</u> : Retourne le premier couple (clé : valeur) d'une carte donnée.....	1377
<u>8.84.11 yp_get_default_domain()</u> : Retourne le domaine NIS par défaut.....	1377
<u>8.84.12 yp_master()</u> : Retourne le nom de la machine maître pour une carte.....	1378
<u>8.84.13 yp_match()</u> : Retourne la ligne associée.....	1378
<u>8.84.14 yp_next()</u> : Retourne le couple (clé : valeur) suivant d'une carte donnée.....	1378
<u>8.84.15 yp_order()</u> : Retourne le numéro d'ordre d'une carte.....	1379
<u>8.85 Lotus Notes</u>	1380
<u>8.85.1 Introduction</u>	1380
<u>8.85.3 notes_copy_db()</u> : Copie une base de données Lotus Notes.....	1380
<u>8.85.4 notes_create_db()</u> : Crée une base Lotus Notes.....	1380
<u>8.85.5 notes_create_note()</u> : Crée une note à partir du formulaire form_name.....	1381
<u>8.85.6 notes_drop_db()</u> : Supprime une base Lotus Notes.....	1381
<u>8.85.7 notes_find_note()</u> : Retourne un identifiant de note, dans le serveur database_name.....	1381
<u>8.85.8 notes_header_info()</u> : Ouvre le message msg_number dans la boîte aux lettres et le serveur spécifiés.....	1381
<u>8.85.9 notes_list_msgs()</u> : Retourne les notes rangées dans database_name.....	1382
<u>8.85.10 notes_mark_read()</u> : Marque la note note_id comme lue par l'utilisateur user_name.....	1382
<u>8.85.11 notes_mark_unread()</u> : Marque la note note_id comme non lue pour l'utilisateur user_name.....	1382
<u>8.85.12 notes_nav_create()</u> : Crée un nom de navigateur dans database_name.....	1383
<u>8.85.13 notes_search()</u> : Cherche les notes par mots-clés dans database_name.....	1383
<u>8.85.14 notes_unread()</u> : Retourne l'identifiant de la note non lue pour l'utilisateur user_name.....	1383
<u>8.85.15 notes_version()</u> : Lit la version de Lotus Notes.....	1383
<u>8.86 Netscape Server API</u>	1385
<u>8.86.1 Introduction</u>	1385
<u>8.86.2 Installation</u>	1385
<u>8.86.3 Configuration à l'exécution</u>	1385
<u>8.86.4 Types de ressources</u>	1385
<u>8.86.5 Constantes pré-définies</u>	1385
<u>8.86.6 Voir aussi</u>	1385
<u>8.86.8 nsapi_response_headers()</u> : Lit tous les en-têtes de réponse HTTP sur Netscape serveur.....	1386
<u>8.86.9 nsapi_virtual()</u> : Effectue une sous-requête NSAPI.....	1386
<u>8.87 agrégation d'objets/Composition de fonctions</u>	1387
<u>8.87.1 Introduction</u>	1387
<u>8.87.2 Exemples</u>	1389
<u>8.87.4 aggregate_methods_by_list()</u> : Agrège sélectivement les méthodes d'une classe grâce à une liste.....	1392
<u>8.87.5 aggregate_methods_by_regexp()</u> : Agrège sélectivement les méthodes d'une classe grâce à une expression rationnelle.....	1392
<u>8.87.6 aggregate_methods()</u> : Agrège dynamiquement les méthodes d'une classe à un objet.....	1393
<u>8.87.7 aggregate_properties_by_list()</u> : Agrège sélectivement les propriétés	

Sommaire

8 Référence des fonctions

<u>d'une classe grâce à une liste</u>	1393
<u>8.87.8 aggregate_properties_by_regexp() : Agrège sélectivement les propriétés d'une classe grâce à une expression rationnelle</u>	1393
<u>8.87.9 aggregate_properties() : Agrège dynamiquement les propriétés d'une classe à un objet</u>	1394
<u>8.87.10 aggregate() : Agrège dynamiquement des classes et objets</u>	1394
<u>8.87.11 aggregation_info() : Alias de aggregate_info</u>	1394
<u>8.87.12 deaggregate() : Desagrège un objet</u>	1394
<u>8.88 Fonctions Oracle</u>	1396
<u>8.88.1 Introduction</u>	1396
<u>8.88.2 Pré-requis</u>	1396
<u>8.88.3 Configuration à l'exécution</u>	1397
<u>8.88.4 Constantes pré-définies</u>	1398
<u>8.88.5 Exemples</u>	1400
<u>8.88.6 Gestion de la connexion</u>	1402
<u>8.88.7 Types de données supportées par le driver</u>	1403
<u>8.88.9 oci_cancel() : Termine la lecture de ressources Oracle</u>	1406
<u>8.88.10 oci_close() : Ferme une connexion Oracle</u>	1406
<u>8.88.11 OCI-Collection->append() : Ajoute un élément à une collection Oracle</u>	1406
<u>8.88.12 OCI-Collection->assign() : Assigne une valeur à une collection depuis une autre collection Oracle</u>	1406
<u>8.88.13 OCI-Collection->assignElem() : Assigne une valeur à un élément d'une collection Oracle</u>	1407
<u>8.88.14 OCI-Collection->free() : Libère les ressources associées avec un objet de collection</u>	1407
<u>8.88.15 OCI-Collection->getElem() : Retourne la valeur d'un élément d'une collection Oracle</u>	1407
<u>8.88.16 OCI-Collection->max() : Retourne le nombre maximal de valeurs d'une collection Oracle</u>	1407
<u>8.88.17 OCI-Collection->size() : Retourne la taille d'une collection Oracle</u>	1407
<u>8.88.18 OCI-Collection->trim() : Supprime les derniers éléments d'une collection Oracle</u>	1408
<u>8.88.19 oci_commit() : Valide les transactions Oracle en cours</u>	1408
<u>8.88.20 oci_connect() : Etablit une connexion avec un serveur Oracle</u>	1409
<u>8.88.21 oci_define_by_name() : Utilise une variable PHP pour la phase de définition, dans une commande SELECT Oracle</u>	1411
<u>8.88.22 oci_error() : Retourne la dernière erreur Oracle</u>	1412
<u>8.88.23 oci_execute() : Exécute une commande SQL Oracle</u>	1413
<u>8.88.24 oci_fetch_all() : Lit toutes les lignes d'un résultat Oracle</u>	1413
<u>8.88.25 oci_fetch_array() : Lit une ligne d'un résultat Oracle sous forme de tableau</u>	1414
<u>8.88.26 oci_fetch_assoc() : Lit une ligne d'un résultat Oracle sous forme de tableau associatif</u>	1416
<u>8.88.27 oci_fetch_object() : Lit une ligne d'un résultat Oracle sous forme d'objet</u>	1416
<u>8.88.28 oci_fetch_row() : Lit une ligne d'un résultat Oracle sous forme de tableau</u>	1417
<u>8.88.29 oci_fetch() : Lit la prochaine ligne dans le résultat Oracle</u>	1417
<u>8.88.30 oci_field_is_null() : Teste si la valeur d'une colonne Oracle est NULL</u>	1417
<u>8.88.31 oci_field_name() : Retourne le nom d'une colonne Oracle</u>	1418
<u>8.88.32 oci_field_precision() : Lit la précision d'une colonne Oracle</u>	1419
<u>8.88.33 oci_field_scale() : Lit l'échelle d'une colonne Oracle</u>	1419
<u>8.88.34 oci_field_size() : Retourne la taille d'un champ Oracle</u>	1419

Sommaire

8 Référence des fonctions

<u>8.88.35 oci_field_type_raw()</u> : Lit directement le type de colonne Oracle.....	1420
<u>8.88.36 oci_field_type()</u> : Retourne le type de données d'un champ Oracle.....	1420
<u>8.88.37 oci_free_statement()</u> : Libère toutes les ressources réservées par un résultat Oracle.....	1421
<u>8.88.38 oci_internal_debug()</u> : Active ou désactive l'affichage des données de débogage Oracle.....	1421
<u>8.88.39 OCI-Lob->append()</u> : Ajoute des données à un LOB Oracle.....	1422
<u>8.88.40 OCI-Lob->close()</u> : Ferme un LOB Oracle.....	1422
<u>8.88.41 oci_lob_copy()</u> : Copie un LOB Oracle.....	1422
<u>8.88.42 OCI-Lob->eof()</u> : Teste la fin du LOB Oracle.....	1422
<u>8.88.43 OCI-Lob->erase()</u> : Ecrase une partie d'un LOB Oracle.....	1422
<u>8.88.44 OCI-Lob->export()</u> : Exporte un LOB Oracle dans un fichier.....	1423
<u>8.88.45 OCI-Lob->flush()</u> : Ecrit les LOB Oracle sur le disque.....	1423
<u>8.88.46 OCI-Lob->free()</u> : Détruit un pointeur de LOB Oracle.....	1423
<u>8.88.47 OCI-Lob->getBuffering()</u> : Retourne l'état de bufferisation LOB d'Oracle.....	1423
<u>8.88.48 OCI-Lob->import()</u> : Importe un fichier dans LOB Oracle.....	1424
<u>8.88.49 oci_lob_is_equal()</u> : Compare deux LOB/FILE Oracle.....	1424
<u>8.88.50 OCI-Lob->load()</u> : Returns large object's contents.....	1424
<u>8.88.51 OCI-Lob->read()</u> : Lit une partie d'un LOB Oracle.....	1424
<u>8.88.52 OCI-Lob->rewind()</u> : Ramène le pointeur interne d'un LOB Oracle au début.....	1424
<u>8.88.53 OCI-Lob->save()</u> : Sauve des données dans un LOB Oracle.....	1424
<u>8.88.54 OCI-Lob->saveFile()</u> : Alias de oci_lob_import.....	1425
<u>8.88.55 OCI-Lob->seek()</u> : Déplace le pointeur interne d'un LOB Oracle.....	1425
<u>8.88.56 OCI-Lob->setBuffering()</u> : Active/désactive la bufferisation des LOB Oracle.....	1425
<u>8.88.57 OCI-Lob->size()</u> : Retourne la taille d'un LOB Oracle.....	1425
<u>8.88.58 OCI-Lob->tell()</u> : Retourne la position courante du pointeur de LOB.....	1425
<u>8.88.59 OCI-Lob->truncate()</u> : Tronque un LOB Oracle.....	1426
<u>8.88.60 OCI-Lob->write()</u> : Ecrit des données dans un LOB Oracle.....	1426
<u>8.88.61 OCI-Lob->writeTemporary()</u> : Ecrit un LOB Oracle temporaire.....	1426
<u>8.88.62 OCI-Lob->writeToFile()</u> : Alias de oci_lob_export.....	1426
<u>8.88.63 oci_new_collection()</u> : Initialise une nouvelle collection Oracle.....	1426
<u>8.88.64 oci_new_connect()</u> : Etablit une nouvelle connexion à Oracle.....	1427
<u>8.88.65 oci_new_cursor()</u> : Alloue un nouveau curseur Oracle.....	1429
<u>8.88.66 oci_new_descriptor()</u> : Initialise un nouveau pointeur vide de LOB/FILE Oracle.....	1430
<u>8.88.67 oci_num_fields()</u> : Retourne le nombre de colonnes dans un résultat Oracle.....	1432
<u>8.88.68 oci_num_rows()</u> : Retourne le nombre de lignes affectées durant la dernière commande Oracle.....	1433
<u>8.88.69 oci_parse()</u> : Prépare une requête SQL avec Oracle.....	1433
<u>8.88.70 oci_password_change()</u> : Modifie le mot de passe d'un utilisateur Oracle.....	1434
<u>8.88.71 oci_pconnect()</u> : Ouvre une connexion persistante à un serveur Oracle.....	1434
<u>8.88.72 oci_result()</u> : Retourne la valeur d'une colonne dans un résultat Oracle.....	1435
<u>8.88.73 oci_rollback()</u> : Annule les transactions Oracle en cours.....	1435
<u>8.88.74 oci_server_version()</u> : Retourne une chaîne contenant les informations de version du serveur Oracle.....	1436
<u>8.88.75 oci_set_prefetch()</u> : Indique le nombre de lignes qui doivent être lues à l'avance par Oracle.....	1436
<u>8.88.76 oci_statement_type()</u> : Retourne le type de commande Oracle.....	1437

Sommaire

8 Référence des fonctions

<u>8.88.77 ocibindbyname()</u> : Alias de <u>oci_bind_by_name</u>	1437
<u>8.88.78 ocicancel()</u> : Alias de <u>oci_cancel</u>	1437
<u>8.88.79 ocicloselob()</u> : Alias de <u>OCI-Lob->close</u>	1437
<u>8.88.80 ocicollappend()</u> : Alias de <u>OCI-Collection->append</u>	1438
<u>8.88.81 ocicollassign()</u> : Alias de <u>OCI-Collection->assign</u>	1438
<u>8.88.82 ocicollassignelem()</u> : Alias de <u>OCI-Collection->assignElem</u>	1438
<u>8.88.83 ocicollgetelem()</u> : Alias de <u>OCI-Collection->getElem</u>	1438
<u>8.88.84 ocicollmax()</u> : Alias de <u>OCI-Collection->max</u>	1438
<u>8.88.85 ocicollsize()</u> : Alias de <u>OCI-Collection->size</u>	1438
<u>8.88.86 ocicolltrim()</u> : Alias de <u>OCI-Collection->trim</u>	1438
<u>8.88.87 ocicolumnisnull()</u> : Alias de <u>oci_field_is_null</u>	1438
<u>8.88.88 ocicolumnname()</u> : Alias de <u>oci_field_name</u>	1438
<u>8.88.89 ocicolumnprecision()</u> : Alias de <u>oci_field_precision</u>	1438
<u>8.88.90 ocicolumnscale()</u> : Alias de <u>oci_field_scale</u>	1439
<u>8.88.91 ocicolumnsize()</u> : Alias de <u>oci_field_size</u>	1439
<u>8.88.92 ocicolumntype()</u> : Alias de <u>oci_field_type</u>	1439
<u>8.88.93 ocicolumntyperaw()</u> : Alias de <u>oci_field_type_raw</u>	1439
<u>8.88.94 ocicommit()</u> : Alias de <u>oci_commit</u>	1439
<u>8.88.95 ocidefinebyname()</u> : Alias de <u>oci_define_by_name</u>	1439
<u>8.88.96 ocierror()</u> : Alias de <u>oci_error</u>	1439
<u>8.88.97 ociexecute()</u> : Alias de <u>oci_execute</u>	1439
<u>8.88.98 ocifetch()</u> : Alias de <u>oci_fetch</u>	1439
<u>8.88.99 ocifetchinto()</u> : Retourne la ligne suivante d'un résultat Oracle dans un tableau.....	1439
<u>8.88.100 ocifetchstatement()</u> : Alias de <u>oci_fetch_all</u>	1440
<u>8.88.101 ocifreecollection()</u> : Alias de <u>OCI-Collection->free</u>	1440
<u>8.88.102 ocifreecursor()</u> : Alias de <u>oci_free_statement</u>	1440
<u>8.88.103 ocifreedesc()</u> : Alias de <u>OCI-Lob->free</u>	1440
<u>8.88.104 ocifreestatement()</u> : Alias de <u>oci_free_statement</u>	1441
<u>8.88.105 ociinternaldebug()</u> : Alias de <u>oci_internal_debug</u>	1441
<u>8.88.106 ociloadlob()</u> : Alias de <u>OCI-Lob->load</u>	1441
<u>8.88.107 ocilogoff()</u> : Alias de <u>oci_close</u>	1441
<u>8.88.108 ocilogon()</u> : Alias de <u>oci_connect</u>	1441
<u>8.88.109 ocinewcollection()</u> : Alias de <u>oci_new_collection</u>	1441
<u>8.88.110 ocinewcursor()</u> : Alias de <u>oci_new_cursor</u>	1441
<u>8.88.111 ocinewscriptor()</u> : Alias de <u>oci_new_descriptor</u>	1441
<u>8.88.112 ocinlogon()</u> : Alias de <u>oci_new_connect</u>	1441
<u>8.88.113 ocinumcols()</u> : Alias de <u>oci_num_fields</u>	1441
<u>8.88.114 ociparse()</u> : Alias de <u>oci_parse</u>	1442
<u>8.88.115 ociplogon()</u> : Alias de <u>oci_pconnect</u>	1442
<u>8.88.116 ocireresult()</u> : Alias de <u>oci_result</u>	1442
<u>8.88.117 ocirollback()</u> : Alias de <u>oci_rollback</u>	1442
<u>8.88.118 ocirowcount()</u> : Alias de <u>oci_num_rows</u>	1442
<u>8.88.119 ocisavelob()</u> : Alias de <u>OCI-Lob->save</u>	1442
<u>8.88.120 ocisavelobfile()</u> : Alias de <u>OCI-Lob->import</u>	1442
<u>8.88.121 ociserverversion()</u> : Alias de <u>oci_server_version</u>	1442
<u>8.88.122 ocisetprefetch()</u> : Alias de <u>oci_set_prefetch</u>	1442
<u>8.88.123 ocistatementtype()</u> : Alias de <u>oci_statement_type</u>	1442
<u>8.88.124 ociwritelobtofile()</u> : Alias de <u>OCI-Lob->export</u>	1443
<u>8.88.125 ociwritetemporarylob()</u> : Alias de <u>OCI-Lob->writeTemporary</u>	1443
<u>8.89 oggvorbis</u>	1444

Sommaire

8 Référence des fonctions

8.89.1 Introduction.....	1444
8.89.2 Pré-requis.....	1444
8.89.3 Configuration à l'exécution.....	1444
8.89.4 Installation.....	1444
8.89.5 Options de contexte.....	1444
8.89.6 Constantes pré-définies.....	1445
8.90 Gestion Audio OpenAL.....	1446
8.90.1 Introduction.....	1446
8.90.2 Installation.....	1446
8.90.3 Configuration à l'exécution.....	1446
8.90.4 Types de ressources.....	1446
8.90.5 Constantes pré-définies.....	1446
8.90.7 <code>openal_buffer_data()</code> : Charge un buffer avec des données.....	1448
8.90.8 <code>openal_buffer_destroy()</code> : Détruit un buffer OpenAL.....	1448
8.90.9 <code>openal_buffer_get()</code> : Récupère les propriétés du buffer OpenAL.....	1448
8.90.10 <code>openal_buffer_loadwav()</code> : Charge un fichier .wav dans le buffer.....	1448
8.90.11 <code>openal_context_create()</code> : Crée un contexte de traitement audio.....	1449
8.90.12 <code>openal_context_current()</code> : Rend courant le contexte spécifié.....	1449
8.90.13 <code>openal_context_destroy()</code> : Détruit un contexte.....	1449
8.90.14 <code>openal_context_process()</code> : Traite le contexte spécifié.....	1449
8.90.15 <code>openal_context_suspend()</code> : Suspend le contexte spécifié.....	1449
8.90.16 <code>openal_device_close()</code> : Ferme un périphérique OpenAL.....	1449
8.90.17 <code>openal_device_open()</code> : Initialise une interface audio OpenAL.....	1449
8.90.18 <code>openal_listener_get()</code> : Récupère une propriété d'auditeur.....	1449
8.90.19 <code>openal_listener_set()</code> : Définit une propriété d'auditeur.....	1449
8.90.20 <code>openal_source_create()</code> : Génère une ressource de source.....	1449
8.90.21 <code>openal_source_destroy()</code> : Détruit une ressource de source.....	1450
8.90.22 <code>openal_source_get()</code> : Récupère une propriété de source OpenAL.....	1450
8.90.23 <code>openal_source_pause()</code> : Marque une pause dans la source.....	1450
8.90.24 <code>openal_source_play()</code> : Démarre la lecture de la source.....	1450
8.90.25 <code>openal_source_rewind()</code> : Revient en arrière dans la source.....	1450
8.90.26 <code>openal_source_set()</code> : Définit une propriété de la source.....	1450
8.90.27 <code>openal_source_stop()</code> : Arrête la lecture de la source.....	1450
8.90.28 <code>openal_stream()</code> : Démarre le streaming d'une source.....	1450
8.91 OpenSSL.....	1451
8.91.1 Introduction.....	1451
8.91.2 Pré-requis.....	1451
8.91.3 Installation.....	1451
8.91.4 Configuration à l'exécution.....	1452
8.91.5 Types de ressources.....	1452
8.91.6 Paramètres clés/certificats.....	1452
8.91.7 Vérification de certificats.....	1452
8.91.8 Constantes pré-définies.....	1452
8.91.10 <code>openssl_csr_export()</code> : Exporte un CSR vers un fichier ou une variable.....	1454
8.91.11 <code>openssl_csr_new()</code> : Génère une CSR.....	1455
8.91.12 <code>openssl_csr_sign()</code> : Signe un CSR avec un autre certificat.....	1456
8.91.13 <code>openssl_error_string()</code> : Retourne le message d'erreur OpenSSL.....	1457
8.91.14 <code>openssl_free_key()</code> : Libère les ressources.....	1458
8.91.15 <code>openssl_get_privatekey()</code> : Alias de <code>openssl_pkey_get_private</code>	1458
8.91.16 <code>openssl_get_publickey()</code> : Alias de <code>openssl_pkey_get_public</code>	1458
8.91.17 <code>openssl_open()</code> : Ouvre des données scellées.....	1458

Sommaire

8 Référence des fonctions

<u>8.91.18 openssl_pkcs7_decrypt()</u> : Déchiffre un message S/MIME.....	1459
<u>8.91.19 openssl_pkcs7_encrypt()</u> : Chiffre un message S/MIME.....	1459
<u>8.91.20 openssl_pkcs7_sign()</u> : Signe un message S/MIME.....	1460
<u>8.91.21 openssl_pkcs7_verify()</u> : Vérifie la signature d'un message S/MIME.....	1461
<u>8.91.22 openssl_pkey_export_to_file()</u> : Sauve une clé au format ASCII dans un fichier.....	1461
<u>8.91.23 openssl_pkey_export()</u> : Lit une représentation exportable de la clé dans une chaîne ou un fichier.....	1461
<u>8.91.24 openssl_pkey_free()</u> : Libère une clé privée.....	1462
<u>8.91.25 openssl_pkey_get_private()</u> : Lit une clé privée.....	1462
<u>8.91.26 openssl_pkey_get_public()</u> : Extrait une clé privée d'un certificat, et la prépare.....	1462
<u>8.91.27 openssl_pkey_new()</u> : Génère une nouvelle clé privée.....	1463
<u>8.91.28 openssl_private_decrypt()</u> : Déchiffre des données avec une clé privée.....	1463
<u>8.91.29 openssl_private_encrypt()</u> : Chiffre des données avec une clé privée.....	1463
<u>8.91.30 openssl_public_decrypt()</u> : Déchiffre des données avec une clé publique.....	1464
<u>8.91.31 openssl_public_encrypt()</u> : Chiffre des données avec une clé publique.....	1464
<u>8.91.32 openssl_seal()</u> : Scelle des données.....	1464
<u>8.91.33 openssl_sign()</u> : Signe les données.....	1465
<u>8.91.34 openssl_verify()</u> : Vérifie une signature.....	1465
<u>8.91.35 openssl_x509_check_private_key()</u> : Vérifie si une clé privée correspond à un certificat.....	1466
<u>8.91.36 openssl_x509_checkpurpose()</u> : Vérifie l'usage d'un certificat.....	1466
<u>8.91.37 openssl_x509_export_to_file()</u> : Exporte un certificat vers un fichier.....	1467
<u>8.91.38 openssl_x509_export()</u> : Exporte un certificat vers une variable.....	1467
<u>8.91.39 openssl_x509_free()</u> : Libère les ressources prises par un certificat.....	1467
<u>8.91.40 openssl_x509_parse()</u> : Analyse un certificat X509.....	1468
<u>8.91.41 openssl_x509_read()</u> : Analyse un certificat X.509 et retourne une ressource.....	1468
<u>8.92 Oracle [obsolète]</u>	1469
<u>8.92.1 Introduction</u>	1469
<u>8.92.2 Installation</u>	1469
<u>8.92.3 Constantes pré-définies</u>	1469
<u>8.92.5 ora_close()</u> : Ferme un pointeur Oracle.....	1470
<u>8.92.6 ora_columnname()</u> : Retourne le nom de la colonne de résultat Oracle.....	1470
<u>8.92.7 ora_columnsize()</u> : Lit la taille d'une colonne Oracle.....	1470
<u>8.92.8 ora_columntype()</u> : Retourne le type de la colonne de résultat Oracle.....	1470
<u>8.92.9 ora_commit()</u> : Valide une transaction Oracle.....	1470
<u>8.92.10 ora_commitoff()</u> : Désactive la validation automatique Oracle.....	1471
<u>8.92.11 ora_commiton()</u> : Active la validation automatique Oracle.....	1471
<u>8.92.12 ora_do()</u> : Analyse, exécute et lit le résultat d'une requête Oracle.....	1471
<u>8.92.13 ora_error()</u> : Retourne le message d'erreur Oracle.....	1471
<u>8.92.14 ora_errorcode()</u> : Retourne le code d'erreur Oracle.....	1472
<u>8.92.15 ora_exec()</u> : Exécute une commande analysée sur un pointeur Oracle.....	1472
<u>8.92.16 ora_fetch_into()</u> : Lit une ligne de résultat Oracle dans un tableau.....	1472
<u>8.92.17 ora_fetch()</u> : Retourne une ligne de résultat.....	1473
<u>8.92.18 ora_getcolumn()</u> : Retourne une donnée d'une ligne de résultat Oracle.....	1473
<u>8.92.19 ora_logoff()</u> : Ferme une connexion Oracle.....	1473
<u>8.92.20 ora_logon()</u> : Ouvre une connexion Oracle.....	1473
<u>8.92.21 ora_numcols()</u> : Retourne le nombre de colonnes.....	1474
<u>8.92.22 ora_numrows()</u> : Retourne le nombre de lignes.....	1474

Sommaire

8 Référence des fonctions

<u>8.92.23 ora_open()</u> : Ouvre un pointeur Oracle.....	1474
<u>8.92.24 ora_parse()</u> : Analyse une requête SQL avec Oracle.....	1474
<u>8.92.25 ora_plogon()</u> : Ouvre une connexion persistante à Oracle.....	1475
<u>8.92.26 ora_rollback()</u> : Annule une transaction Oracle.....	1475
<u>8.93 Bufferisation de sortie</u>	1476
<u>8.93.1 Introduction</u>	1476
<u>8.93.2 Pré-requis</u>	1476
<u>8.93.3 Installation</u>	1476
<u>8.93.4 Configuration à l'exécution</u>	1476
<u>8.93.5 Types de ressources</u>	1477
<u>8.93.6 Constantes pré-définies</u>	1477
<u>8.93.7 Exemples</u>	1477
<u>8.93.8 Voir aussi</u>	1477
<u>8.93.10 ob_clean()</u> : Efface le buffer de sortie.....	1478
<u>8.93.11 ob_end_clean()</u> : Détruit les données du buffer de sortie et éteint la bufferisation de sortie.....	1478
<u>8.93.12 ob_end_flush()</u> : Envoie les données du buffer de sortie et éteint la bufferisation de sortie.....	1479
<u>8.93.13 ob_flush()</u> : Envoie le buffer de sortie.....	1479
<u>8.93.14 ob_get_clean()</u> : Lit le contenu courant du buffer de sortie puis l'efface.....	1479
<u>8.93.15 ob_get_contents()</u> : Retourne le contenu du buffer de sortie.....	1480
<u>8.93.16 ob_get_flush()</u> : Vide le tampon, le retourne en tant que chaîne et stoppe la mise en tampon.....	1480
<u>8.93.17 ob_get_length()</u> : Retourne la longueur du contenu du buffer de sortie.....	1481
<u>8.93.18 ob_get_level()</u> : Retourne le nombre de niveaux d'imbrications du système de bufferisation de sortie.....	1481
<u>8.93.19 ob_get_status()</u> : Lit le statut du buffer de sortie.....	1481
<u>8.93.20 ob_gzhandler()</u> : Fonction de callback pour la compression automatique des buffers.....	1481
<u>8.93.21 ob_implicit_flush()</u> : Active/désactive l'envoi implicite.....	1482
<u>8.93.22 ob_list_handlers()</u> : Liste les gestionnaires d'affichage utilisés.....	1482
<u>8.93.23 ob_start()</u> : Enclenche la bufferisation de sortie.....	1483
<u>8.93.24 output_add_rewrite_var()</u> : Ajoute une règle de réécriture d'URL.....	1485
<u>8.93.25 output_reset_rewrite_vars()</u> : Annule la réécriture d'URL.....	1485
<u>8.94 Overload</u>	1487
<u>8.94.1 Introduction</u>	1487
<u>8.94.2 Pré-requis</u>	1487
<u>8.94.3 Installation</u>	1487
<u>8.94.4 Configuration à l'exécution</u>	1487
<u>8.94.5 Types de ressources</u>	1487
<u>8.94.6 Constantes pré-définies</u>	1487
<u>8.94.7 Exemples</u>	1488
<u>8.95 Ovrimos SQL</u>	1489
<u>8.95.1 Introduction</u>	1489
<u>8.95.2 Pré-requis</u>	1489
<u>8.95.3 Installation</u>	1489
<u>8.95.4 Configuration à l'exécution</u>	1489
<u>8.95.5 Types de ressources</u>	1489
<u>8.95.6 Exemples</u>	1489
<u>8.95.8 ovrimos_commit()</u> : Valide une transaction Ovrimos.....	1490
<u>8.95.9 ovrimos_connect()</u> : Connexion à un serveur.....	1490

Sommaire

8 Référence des fonctions

<u>8.95.10 ovrimos_cursor()</u> : Retourne le nom du curseur Ovrimos.....	1491
<u>8.95.11 ovrimos_exec()</u> : Exécute une requête SQL.....	1491
<u>8.95.12 ovrimos_execute()</u> : Exécute une requête Ovrimos préparée.....	1491
<u>8.95.13 ovrimos_fetch_into()</u> : Lit une ligne dans un résultat Ovrimos.....	1491
<u>8.95.14 ovrimos_fetch_row()</u> : Lit une ligne dans un résultat Ovrimos.....	1492
<u>8.95.15 ovrimos_field_len()</u> : Retourne la taille d'une colonne Ovrimos.....	1493
<u>8.95.16 ovrimos_field_name()</u> : Retourne le nom d'une colonne Ovrimos.....	1493
<u>8.95.17 ovrimos_field_num()</u> : Retourne le numéro de colonne Ovrimos.....	1493
<u>8.95.18 ovrimos_field_type()</u> : Retourne le type numérique d'une colonne Ovrimos.....	1493
<u>8.95.19 ovrimos_free_result()</u> : Libère les ressources utilisées par un résultat Ovrimos.....	1493
<u>8.95.20 ovrimos_longreadlen()</u> : Indique la taille des données à lire dans une colonne de grande taille.....	1493
<u>8.95.21 ovrimos_num_fields()</u> : Retourne le nombre de colonnes Ovrimos.....	1494
<u>8.95.22 ovrimos_num_rows()</u> : Retourne le nombre de lignes affectées par une modification Ovrimos.....	1494
<u>8.95.23 ovrimos_prepare()</u> : Prépare une requête SQL Ovrimos.....	1494
<u>8.95.24 ovrimos_result_all()</u> : Affiche un résultat Ovrimos sous forme de table HTML.....	1494
<u>8.95.25 ovrimos_result()</u> : Lit le contenu d'une colonne Ovrimos.....	1496
<u>8.95.26 ovrimos_rollback()</u> : Annule une transaction.....	1496
<u>8.96 Accès aux fichiers Paradox</u>	1497
<u>8.96.1 Introduction</u>	1497
<u>8.96.2 Pré-requis</u>	1497
<u>8.96.3 Installation</u>	1497
<u>8.96.4 Configuration à l'exécution</u>	1497
<u>8.96.5 API orientée objet</u>	1497
<u>8.96.6 Constantes pré-définies</u>	1498
<u>8.96.8 px_create_fp()</u> : Crée une nouvelle base de données Paradox.....	1499
<u>8.96.9 px_delete()</u> : Efface la ressource d'une base de données Paradox.....	1500
<u>8.96.10 px_get_field()</u> : Retourne les spécifications d'un champ.....	1500
<u>8.96.11 px_get_info()</u> : Retourne des informations sur un fichier Paradox.....	1500
<u>8.96.12 px_get_parameter()</u> : Récupère un paramètre.....	1500
<u>8.96.13 px_get_record()</u> : Retourne un enregistrement d'une base de données Paradox.....	1500
<u>8.96.14 px_get_schema()</u> : Retourne le schéma de la base de données.....	1500
<u>8.96.15 px_get_value()</u> : Récupère une valeur.....	1500
<u>8.96.16 px_new()</u> : Crée un nouvel objet Paradox.....	1501
<u>8.96.17 px_numfields()</u> : Retourne le nombre de champs d'une base de données Paradox.....	1501
<u>8.96.18 px_numrecords()</u> : Retourne le nombre d'enregistrements d'une base de données Paradox.....	1501
<u>8.96.19 px_open_fp()</u> : Ouvre une base de données Paradox.....	1501
<u>8.96.20 px_put_record()</u> : Stock un enregistrement dans une base de données Paradox.....	1501
<u>8.96.21 px_set_blob_file()</u> : Définit le fichier depuis lequel les blobs seront lus.....	1501
<u>8.96.22 px_set_parameter()</u> : Définit un paramètre.....	1502
<u>8.96.23 px_set_tablename()</u> : Définit le nom de la table (déprécié).....	1502
<u>8.96.24 px_set_targetencoding()</u> : Définit l'encodage pour les caractères des champs (déprécié).....	1502

Sommaire

8 Référence des fonctions

<u>8.96.25 px_set_value()</u> : Définit une valeur.....	1502
<u>8.96.26 px_timestamp2string()</u> : Convertit un timestamp en une chaîne de caractères.....	1502
8.97 Fonctions Parsekit	1503
<u>8.97.1 Introduction</u>	1503
<u>8.97.2 Installation</u>	1503
<u>8.97.3 Configuration à l'exécution</u>	1503
<u>8.97.4 Types de ressources</u>	1503
<u>8.97.5 Constantes pré-définies</u>	1503
<u>8.97.7 parsekit_compile_string()</u> : Compile une chaîne de code PHP et retourne le tableau d'opcode résultant.....	1509
<u>8.97.8 parsekit_func_arginfo()</u> : Retourne des informations concernant les arguments d'une fonction.....	1509
8.98 Contrôle des processus	1510
<u>8.98.1 Introduction</u>	1510
<u>8.98.2 Pré-requis</u>	1510
<u>8.98.3 Installation</u>	1510
<u>8.98.4 Configuration à l'exécution</u>	1510
<u>8.98.5 Types de ressources</u>	1510
<u>8.98.6 Constantes pré-définies</u>	1511
<u>8.98.7 Exemples</u>	1511
<u>8.98.8 Voir aussi</u>	1512
<u>8.98.10 pcntl_exec()</u> : Exécute le programme indiqué dans l'espace courant de processus.....	1513
<u>8.98.11 pcntl_fork()</u> : Duplique le process courant.....	1513
<u>8.98.12 pcntl_getpriority()</u> : Retourne la priorité d'un processus.....	1514
<u>8.98.13 pcntl_setpriority()</u> : Change la priorité d'un processus.....	1514
<u>8.98.14 pcntl_signal()</u> : Installe un gestionnaire de signaux.....	1514
<u>8.98.15 pcntl_wait()</u> : Attend ou retourne le statut d'un processus fils.....	1516
<u>8.98.16 pcntl_waitpid()</u> : Attend la fin de l'exécution d'un processus fils.....	1516
<u>8.98.17 pcntl_wexitstatus()</u> : Retourne le code d'un processus fils terminé.....	1517
<u>8.98.18 pcntl_wifexited()</u> : Retourne TRUE si le code de retour représente une fin normale.....	1517
<u>8.98.19 pcntl_wifsignaled()</u> : Retourne TRUE si le code statut représente une fin due à un signal.....	1517
<u>8.98.20 pcntl_wifstopped()</u> : Retourne TRUE si le processus fils est stoppé.....	1518
<u>8.98.21 pcntl_wstopsig()</u> : Retourne le signal qui a causé l'arrêt du processus fils.....	1518
<u>8.98.22 pcntl_wtermsig()</u> : Retourne le signal qui a provoqué la fin du processus fils.....	1518
8.99 Expressions rationnelles compatibles Perl	1519
<u>8.99.1 Introduction</u>	1519
<u>8.99.2 Pré-requis</u>	1519
<u>8.99.3 Installation</u>	1519
<u>8.99.4 Configuration à l'exécution</u>	1519
<u>8.99.5 Types de ressources</u>	1519
<u>8.99.6 Constantes pré-définies</u>	1520
<u>8.99.7 Exemples</u>	1520
<u>8.99.9 syntaxe des masques()</u> : Fonctionnement des expressions rationnelles.....	1521
<u>8.99.10 preg_grep()</u> : Retourne un tableau avec les résultats de la recherche.....	1521
<u>8.99.11 preg_match_all()</u> : Expression rationnelle globale.....	1521
<u>8.99.12 preg_match()</u> : Expression rationnelle standard.....	1524

Sommaire

8 Référence des fonctions

<u>8.99.13 preg_quote()</u> : Echappement des caractères spéciaux des expressions rationnelles.....	1526
<u>8.99.14 preg_replace_callback()</u> : Rechercher/remplacer avec une expression rationnelle et fonction de callback.....	1526
<u>8.99.15 preg_replace()</u> : Rechercher et remplacer par expression rationnelle standard.....	1527
<u>8.99.16 preg_split()</u> : Eclate une chaîne par expression rationnelle.....	1528
<u>8.100 PDF</u>	1530
<u>8.100.1 Introduction</u>	1530
<u>8.100.2 Pré-requis</u>	1530
<u>8.100.3 Installation</u>	1530
<u>8.100.4 Configuration à l'exécution</u>	1531
<u>8.100.5 Confusion entre les vieilles versions de PDFLib</u>	1531
<u>8.100.6 Exemples</u>	1532
<u>8.100.7 Voir aussi</u>	1534
<u>8.100.9 pdf_add_bookmark()</u> : Ajoute un signet dans la page courante.....	1537
<u>8.100.10 pdf_add_launchlink()</u> : Ajoute une annotation de lancement dans la page PDF courante.....	1537
<u>8.100.11 pdf_add_loclink()</u> : Ajoute une annotation de lien dans la page PDF courante.....	1538
<u>8.100.12 pdf_add_note()</u> : Ajoute une annotation dans la page PDF courante.....	1538
<u>8.100.13 pdf_add_outline()</u> : Obsolète : ajoute un signet dans la page courante.....	1538
<u>8.100.14 pdf_add_pdflink()</u> : Ajoute une annotation sur un lien vers un fichier dans la page PDF courante.....	1539
<u>8.100.15 pdf_add_thumbnail()</u> : Ajoute une miniature sur la page PDF courante.....	1539
<u>8.100.16 pdf_add_weblink()</u> : Ajoute un lien web sur la page PDF courante.....	1539
<u>8.100.17 pdf_arc()</u> : Dessine un arc de cercle PDF dans le sens antihoraire.....	1539
<u>8.100.18 pdf_arcn()</u> : Dessine un arc de cercle dans le sens horaire.....	1540
<u>8.100.19 pdf_attach_file()</u> : Ajoute un fichier attaché à la page PDF.....	1541
<u>8.100.20 pdf_begin_page()</u> : Initialise une nouvelle page de document PDF.....	1541
<u>8.100.21 pdf_begin_pattern()</u> : Initialise un nouveau pattern PDF.....	1541
<u>8.100.22 pdf_begin_template()</u> : Initialise un nouveau template PDF.....	1542
<u>8.100.23 pdf_circle()</u> : Dessine un cercle dans un document PDF.....	1542
<u>8.100.24 pdf_clip()</u> : Modifie le chemin de clipping PDF.....	1542
<u>8.100.25 pdf_close_image()</u> : Ferme une image dans un document PDF.....	1543
<u>8.100.26 pdf_close_pdi_page()</u> : Ferme la page PDF.....	1543
<u>8.100.27 pdf_close_pdi()</u> : Ferme le fichier PDF d'entrée.....	1543
<u>8.100.28 pdf_close()</u> : Ferme le fichier PDF.....	1543
<u>8.100.29 pdf_closepath_fill_stroke()</u> : Termine le chemin, dessine les bords et remplit la forme.....	1543
<u>8.100.30 pdf_closepath_stroke()</u> : Termine le chemin et dessine les bords.....	1543
<u>8.100.31 pdf_closepath()</u> : Termine le chemin PDF courant.....	1544
<u>8.100.32 pdf_concat()</u> : Concatène une matrice avec le CTM.....	1544
<u>8.100.33 pdf_continue_text()</u> : Affiche du texte à la prochaine ligne PDF.....	1544
<u>8.100.34 pdf_curveto()</u> : Dessine une courbe de Bezier.....	1544
<u>8.100.35 pdf_delete()</u> : Efface un objet PDF.....	1544
<u>8.100.36 pdf_end_page()</u> : Termine la page PDF courante.....	1545
<u>8.100.37 pdf_end_pattern()</u> : Termine le pattern PDF.....	1545
<u>8.100.38 pdf_end_template()</u> : Termine le template PDF.....	1545
<u>8.100.39 pdf_endpath()</u> : Obsolète : termine le chemin courant.....	1545
<u>8.100.40 pdf_fill_stroke()</u> : Remplit et passe le pinceau sur le chemin PDF	

Sommaire

8 Référence des fonctions

<u>courant</u>	1545
<u>8.100.41 pdf_fill()</u> : Remplit le chemin PDF courant avec la couleur courante.....	1545
<u>8.100.42 pdf_findfont()</u> : Prépare une police pour utilisation ultérieure avec <u>pdf_setfont</u>	1546
<u>8.100.43 pdf_get_buffer()</u> : Lit le tampon contenant le fichier PDF généré.....	1546
<u>8.100.44 pdf_get_font()</u> : Obsolète: charge une police.....	1546
<u>8.100.45 pdf_get_fontname()</u> : Obsolète : lit le nom de la police.....	1546
<u>8.100.46 pdf_get_fontsize()</u> : Obsolète : gère les polices.....	1547
<u>8.100.47 pdf_get_image_height()</u> : Obsolète : retourne la hauteur d'une image.....	1547
<u>8.100.48 pdf_get_image_width()</u> : Obsolète : retourne la largeur d'une image.....	1547
<u>8.100.49 pdf_get_majorversion()</u> : Retourne le numéro de version majeur de <u>PDFlib</u>	1547
<u>8.100.50 pdf_get_minorversion()</u> : Retourne le numéro de version mineure de <u>PDFlib</u>	1547
<u>8.100.51 pdf_get_parameter()</u> : Lit certains paramètres.....	1547
<u>8.100.52 pdf_get_pdi_parameter()</u> : Lit des paramètres textuels dans le <u>document PDI</u>	1548
<u>8.100.53 pdf_get_pdi_value()</u> : Lit des paramètres numériques dans le <u>document PDF d'entrée</u>	1548
<u>8.100.54 pdf_get_value()</u> : Lit certains paramètres numériques.....	1548
<u>8.100.55 pdf_initgraphics()</u> : Remet à zéro l'environnement graphique PDF.....	1548
<u>8.100.56 pdf_lineto()</u> : Dessine une ligne PDF.....	1548
<u>8.100.57 pdf_makespotcolor()</u> : Place un point de couleur PDF.....	1548
<u>8.100.58 pdf_moveto()</u> : Place le point courant PDF.....	1549
<u>8.100.59 pdf_new()</u> : Crée un nouvel objet PDF.....	1549
<u>8.100.60 pdf_open_ccitt()</u> : Ouvre une image contenant des données brutes <u>CCITT</u>	1549
<u>8.100.61 pdf_open_file()</u> : Ouvre un nouveau fichier PDF.....	1549
<u>8.100.62 pdf_open_gif()</u> : Obsolète : ouvre une image GIF.....	1550
<u>8.100.63 pdf_open_image_file()</u> : Lit une image depuis un fichier.....	1550
<u>8.100.64 pdf_open_image()</u> : Ouvre une image.....	1550
<u>8.100.65 pdf_open_jpeg()</u> : Obsolète : ouvre une image JPEG.....	1550
<u>8.100.66 pdf_open_memory_image()</u> : Ouvre une image créée en mémoire par <u>PHP</u>	1550
<u>8.100.67 pdf_open_pdi_page()</u> : Prépare une page.....	1551
<u>8.100.68 pdf_open_pdi()</u> : Ouvre un fichier PDF.....	1551
<u>8.100.69 pdf_open_png()</u> : Obsolète : ouvre une image PNG.....	1551
<u>8.100.70 pdf_open_tiff()</u> : Obsolète : ouvre une image TIFF.....	1551
<u>8.100.71 pdf_open()</u> : Obsolète : ouvre un nouvel objet PDF.....	1551
<u>8.100.72 pdf_place_image()</u> : Place une image dans la page.....	1552
<u>8.100.73 pdf_place_pdi_page()</u> : Place une page dans le document.....	1552
<u>8.100.74 pdf_rect()</u> : Dessine un rectangle.....	1552
<u>8.100.75 pdf_restore()</u> : Rétablit l'ancien environnement graphique PDF.....	1552
<u>8.100.76 pdf_rotate()</u> : Configure la rotation.....	1552
<u>8.100.77 pdf_save()</u> : Sauve l'environnement graphique courant.....	1552
<u>8.100.78 pdf_scale()</u> : Configure l'échelle du document.....	1552
<u>8.100.79 pdf_set_border_color()</u> : Configure la couleur des bords autour des <u>liens et annotations</u>	1553
<u>8.100.80 pdf_set_border_dash()</u> : Configure le style des lignes autour des <u>liens et annotations</u>	1553
<u>8.100.81 pdf_set_border_style()</u> : Choisit le style de bord autour des <u>liens et annotations</u>	1553

Sommaire

8 Référence des fonctions

<u>annotations</u>	1553
<u>8.100.82 pdf_set_char_spacing()</u> : Obsolète : configure l'espacement de caractères.....	1553
<u>8.100.83 pdf_set_duration()</u> : Obsolète : configure la durée entre deux pages.....	1553
<u>8.100.84 pdf_set_font()</u> : Obsolète : sélectionne la taille et le nom d'une police.....	1554
<u>8.100.85 pdf_set_horiz_scaling()</u> : Obsolète : configure l'échelle horizontale du texte.....	1554
<u>8.100.86 pdf_set_info_author()</u> : Obsolète : remplit le champ d'auteur du document.....	1554
<u>8.100.87 pdf_set_info_creator()</u> : Obsolète : remplit le champ de créateur du document.....	1554
<u>8.100.88 pdf_set_info_keywords()</u> : Obsolète : remplit le champ de mots-clés du document.....	1554
<u>8.100.89 pdf_set_info_subject()</u> : Obsolète : remplit le champ de sujet du document.....	1554
<u>8.100.90 pdf_set_info_title()</u> : Obsolète : remplit le champ de titre du document.....	1555
<u>8.100.91 pdf_set_info()</u> : Remplit un champ de l'en-tête de document PDF.....	1555
<u>8.100.92 pdf_set_leading()</u> : Obsolète : configure la distance entre deux lignes de texte.....	1555
<u>8.100.93 pdf_set_parameter()</u> : Modifie certains paramètres.....	1555
<u>8.100.94 pdf_set_text_matrix()</u> : Obsolète : configure la matrice de texte.....	1555
<u>8.100.95 pdf_set_text_pos()</u> : Modifie la position du texte.....	1555
<u>8.100.96 pdf_set_text_rendering()</u> : Obsolète : détermine le rendu du texte.....	1556
<u>8.100.97 pdf_set_text_rise()</u> : Obsolète : configure l'élévation de texte.....	1556
<u>8.100.98 pdf_set_value()</u> : Modifie certains paramètres numériques.....	1556
<u>8.100.99 pdf_set_word_spacing()</u> : Obsolète : configure l'espace entre deux mots.....	1556
<u>8.100.100 pdf_setcolor()</u> : Configure la couleur de dessin et de remplissage.....	1556
<u>8.100.101 pdf_setdash()</u> : Configure le mode de pointillé.....	1557
<u>8.100.102 pdf_setflat()</u> : Configure la position à plat (flatness).....	1557
<u>8.100.103 pdf_setfont()</u> : Configure la police courante.....	1557
<u>8.100.104 pdf_setgray_fill()</u> : Configure la couleur de remplissage à un niveau de gris.....	1558
<u>8.100.105 pdf_setgray_stroke()</u> : Configure la couleur de dessin à un niveau de gris.....	1558
<u>8.100.106 pdf_setgray()</u> : Configure la couleur de dessin et de remplissage à un niveau de gris.....	1558
<u>8.100.107 pdf_setlinecap()</u> : Configure le paramètre de linecap.....	1558
<u>8.100.108 pdf_setlinejoin()</u> : Configure le paramètre de linejoin.....	1558
<u>8.100.109 pdf_setlinewidth()</u> : Configure la largeur de ligne.....	1559
<u>8.100.110 pdf_setmatrix()</u> : Configure la matrice de transformation.....	1559
<u>8.100.111 pdf_setmiterlimit()</u> : Configure la "miter limit".....	1559
<u>8.100.112 pdf_setpolydash()</u> : Obsolète : configure des pointillés complexes.....	1559
<u>8.100.113 pdf_setrgbcolor_fill()</u> : Choisit la couleur utilisée pour le remplissage.....	1559
<u>8.100.114 pdf_setrgbcolor_stroke()</u> : Choisit la couleur utilisée pour le dessin.....	1559
<u>8.100.115 pdf_setrgbcolor()</u> : Choisit la couleur rgb de remplissage et de dessin.....	1560
<u>8.100.116 pdf_show_boxed()</u> : Affiche le texte dans un cadre.....	1560
<u>8.100.117 pdf_show_xy()</u> : Affiche un texte à une position donnée.....	1560
<u>8.100.118 pdf_show()</u> : Affiche le texte à la position courante.....	1560
<u>8.100.119 pdf_skew()</u> : Incline le système de coordonnées.....	1561
<u>8.100.120 pdf_stringwidth()</u> : Retourne la largeur d'un texte avec la police courante.....	1561

Sommaire

8 Référence des fonctions

<u>8.100.121 pdf_stroke() : Dessine la ligne le long du chemin</u>	1561
<u>8.100.122 pdf_translate() : Effectue une translation de l'origine du système de coordonnées</u>	1561
8.101 Fonctions PDO	1562
<u>8.101.1 Introduction</u>	1562
<u>8.101.2 Installation</u>	1562
<u>8.101.3 Configuration à l'exécution</u>	1563
<u>8.101.4 Drivers PDO</u>	1564
<u>8.101.5 Connexions et gestionnaire de connexion</u>	1564
<u>8.101.6 Transactions et validation automatique (auto-commit)</u>	1565
<u>8.101.7 Requêtes préparées et procédures stockées</u>	1566
<u>8.101.8 Erreurs et gestionnaire des erreurs</u>	1568
<u>8.101.9 Les objets larges (LOBs)</u>	1569
<u>8.101.10 Classes pré-définies</u>	1571
<u>8.101.11 Constantes pré-définies</u>	1572
<u>8.101.13 PDO::commit() : Valide une transaction</u>	1576
<u>8.101.14 PDO::__construct() : Crée une instance PDO qui représente une connexion à une base de données</u>	1576
<u>8.101.15 PDO::errorCode() : Retourne le SQLSTATE associé avec la dernière opération sur la base de données</u>	1576
<u>8.101.16 PDO::errorInfo() : Retourne les informations associées à l'erreur lors de la dernière opération sur la base de données</u>	1577
<u>8.101.17 PDO::exec() : Exécute une requête SQL et retourne le nombre de lignes affectées</u>	1577
<u>8.101.18 PDO::getAttribute() : Récupère un attribut d'une connexion à une base de données</u>	1577
<u>8.101.19 PDO::getAvailableDrivers() : Retourne un tableau des pilotes PDO disponibles</u>	1577
<u>8.101.20 PDO::lastInsertId() : Retourne l'identifiant de la dernière ligne insérée ou la valeur d'une séquence</u>	1577
<u>8.101.21 PDO::prepare() : Prépare une requête à l'exécution et retourne un objet</u>	1578
<u>8.101.22 PDO::query() : Exécute une requête SQL, retourne un jeu de résultats en tant qu'objet PDOStatement</u>	1578
<u>8.101.23 PDO::quote() : Protège une chaîne pour l'utiliser dans une requête SQL PDO</u>	1578
<u>8.101.24 PDO::rollback() : Annule une transaction</u>	1579
<u>8.101.25 PDO::setAttribute() : Définit un attribut</u>	1579
<u>8.101.26 PDOStatement::bindColumn() : Lie une colonne à une variable PHP</u>	1580
<u>8.101.27 PDOStatement::bindParam() : Lie un paramètre à un nom de variable spécifique</u>	1580
<u>8.101.28 PDOStatement::bindValue() : Associe une valeur à un paramètre</u>	1580
<u>8.101.29 PDOStatement::closeCursor() : Ferme le curseur, permettant à la requête d'être de nouveau exécutée</u>	1580
<u>8.101.30 PDOStatement::columnCount() : Retourne le nombre de colonnes dans le jeu de résultats</u>	1581
<u>8.101.31 PDOStatement::errorCode() : Récupère le SQLSTATE associé lors de la dernière opération sur la requête</u>	1581
<u>8.101.32 PDOStatement::errorInfo() : Récupère les informations sur l'erreur associée lors dernière opération sur la requête</u>	1581
<u>8.101.33 PDOStatement::execute() : Exécute une requête préparée</u>	1582

Sommaire

8 Référence des fonctions

<u>8.101.34 PDOStatement::fetch() : Récupère la ligne suivante d'un jeu de résultat PDO.....</u>	1582
<u>8.101.35 PDOStatement::fetchAll() : Retourne un tableau contenant toutes les lignes du jeu d'enregistrements.....</u>	1582
<u>8.101.36 PDOStatement::fetchColumn() : Retourne une colonne depuis la ligne suivante d'un jeu de résultats.....</u>	1582
<u>8.101.37 PDOStatement::getAttribute() : Récupère un attribut de requête.....</u>	1582
<u>8.101.38 PDOStatement::getColumnMeta() : Retourne les méta-données pour une colonne d'un jeu de résultats.....</u>	1582
<u>8.101.39 PDOStatement::nextRowset() : Avance à la prochaine ligne de résultats d'un gestionnaire de lignes de résultats multiples.....</u>	1583
<u>8.101.40 PDOStatement::rowCount() : Retourne le nombre de lignes affectées par le dernier appel à la fonction PDOStatement::execute().....</u>	1583
<u>8.101.41 PDOStatement::setAttribute() : Définit un attribut de requête.....</u>	1583
<u>8.101.42 PDOStatement::setFetchMode() : Définit le mode de récupération par défaut pour cette requête.....</u>	1583
<u>8.102 Paiement par Verisign.....</u>	1584
<u>8.102.1 Introduction.....</u>	1584
<u>8.102.2 Pré-requis.....</u>	1584
<u>8.102.3 Installation.....</u>	1584
<u>8.102.4 Configuration à l'exécution.....</u>	1585
<u>8.102.5 Types de ressources.....</u>	1585
<u>8.102.6 Constantes pré-définies.....</u>	1585
<u>8.102.8 pfpro_init() : Initialise la bibliothèque Payflow Pro.....</u>	1586
<u>8.102.9 pfpro_process_raw() : Envoie une transaction brute à Payflow Pro.....</u>	1586
<u>8.102.10 pfpro_process() : Effectue une transaction avec Payflow Pro.....</u>	1586
<u>8.102.11 pfpro_version() : Retourne le numéro de version de Payflow Pro.....</u>	1588
<u>8.103 PostgreSQL.....</u>	1589
<u>8.103.1 Introduction.....</u>	1589
<u>8.103.2 Pré-requis.....</u>	1589
<u>8.103.3 Installation.....</u>	1589
<u>8.103.4 Configuration à l'exécution.....</u>	1589
<u>8.103.5 Types de ressources.....</u>	1590
<u>8.103.6 Constantes pré-définies.....</u>	1590
<u>8.103.7 Notes.....</u>	1593
<u>8.103.8 Exemples.....</u>	1593
<u>8.103.10 pg_cancel_query() : Annule une requête asynchrone.....</u>	1595
<u>8.103.11 pg_client_encoding() : Lit l'encodage du client.....</u>	1595
<u>8.103.12 pg_close() : Termine une connexion PostgreSQL.....</u>	1596
<u>8.103.13 pg_connect() : Établit une connexion PostgreSQL.....</u>	1596
<u>8.103.14 pg_connection_busy() : Vérifie si la connexion PostgreSQL est occupée.....</u>	1596
<u>8.103.15 pg_connection_reset() : Relance la connexion au serveur PostgreSQL.....</u>	1596
<u>8.103.16 pg_connection_status() : Lit le statut de la connexion PostgreSQL.....</u>	1597
<u>8.103.17 pg_convert() : Convertit des tableaux associatifs en une commande PostgreSQL.....</u>	1597
<u>8.103.18 pg_copy_from() : Insère des lignes dans une table à partir d'un tableau.....</u>	1597
<u>8.103.19 pg_copy_to() : Copie une table dans un tableau.....</u>	1597
<u>8.103.20 pg_dbname() : Retourne le nom de la base de données PostgreSQL.....</u>	1597
<u>8.103.21 pg_delete() : Efface des lignes PostgreSQL.....</u>	1598
<u>8.103.22 pg_end_copy() : Synchronise avec le serveur PostgreSQL.....</u>	1598

Sommaire

8 Référence des fonctions

<u>8.103.23 pg_escape_bytea() : Protège une chaîne pour insertion dans un champ bytea.....</u>	1598
<u>8.103.24 pg_escape_string() : Protège une chaîne de caractères pour l'insérer dans un champ texte.....</u>	1598
<u>8.103.25 pg_execute() : Envoie une requête pour exécuter une requête préparée avec les paramètres donnés et attend le résultat.....</u>	1599
<u>8.103.26 pg_fetch_all_columns() : Récupère toutes les lignes d'une colonne de résultats particulière en tant que tableau.....</u>	1599
<u>8.103.27 pg_fetch_all() : Lit toutes les lignes d'un résultat.....</u>	1599
<u>8.103.28 pg_fetch_array() : Lit une ligne de résultat PostgreSQL dans un tableau.....</u>	1599
<u>8.103.29 pg_fetch_assoc() : Lit une ligne de résultat PostgreSQL sous forme de tableau numérique.....</u>	1600
<u>8.103.30 pg_fetch_object() : Lit une ligne de résultat PostgreSQL dans un objet.....</u>	1600
<u>8.103.31 pg_fetch_result() : Retourne les valeurs d'un résultat.....</u>	1601
<u>8.103.32 pg_fetch_row() : Lit une ligne dans un tableau.....</u>	1601
<u>8.103.33 pg_field_is_null() : Teste si un champ PostgreSQL est à NULL.....</u>	1601
<u>8.103.34 pg_field_name() : Retourne le nom d'un champ PostgreSQL.....</u>	1601
<u>8.103.35 pg_field_num() : Retourne le numéro d'une colonne.....</u>	1601
<u>8.103.36 pg_field_prtlen() : Retourne la taille imprimée.....</u>	1602
<u>8.103.37 pg_field_size() : Retourne la taille interne de stockage d'un champ donné.....</u>	1602
<u>8.103.38 pg_field_type_oid() : Retourne le type ID (OID) pour le numéro du champ correspondant.....</u>	1602
<u>8.103.39 pg_field_type() : Retourne le type d'un champ PostgreSQL donné par index.....</u>	1603
<u>8.103.40 pg_free_result() : Libère la mémoire.....</u>	1603
<u>8.103.41 pg_get_notify() : Lit le message SQL NOTIFY.....</u>	1603
<u>8.103.42 pg_get_pid() : Lit l'identifiant de processus du serveur PostgreSQL.....</u>	1603
<u>8.103.43 pg_get_result() : Lit un résultat PostgreSQL asynchrone.....</u>	1603
<u>8.103.44 pg_host() : Retourne le nom d'hôte.....</u>	1604
<u>8.103.45 pg_insert() : Insère un tableau dans une table.....</u>	1604
<u>8.103.46 pg_last_error() : Lit le dernier message d'erreur sur la connexion.....</u>	1604
<u>8.103.47 pg_last_notice() : Retourne la dernière note du serveur PostgreSQL.....</u>	1604
<u>8.103.48 pg_last_oid() : Retourne l'identifiant de la dernière ligne.....</u>	1605
<u>8.103.49 pg_lo_close() : Ferme un objet de grande taille PostgreSQL.....</u>	1605
<u>8.103.50 pg_lo_create() : Crée un objet de grande taille PostgreSQL.....</u>	1605
<u>8.103.51 pg_lo_export() : Exporte un objet de grande taille vers un fichier.....</u>	1606
<u>8.103.52 pg_lo_import() : Importe un objet de grande taille depuis un fichier.....</u>	1606
<u>8.103.53 pg_lo_open() : Ouvre un objet de grande taille PostgreSQL.....</u>	1607
<u>8.103.54 pg_lo_read_all() : Lit un objet de grande taille en totalité.....</u>	1607
<u>8.103.55 pg_lo_read() : Lit un objet de grande taille.....</u>	1607
<u>8.103.56 pg_lo_seek() : Modifie la position dans un objet de grande taille.....</u>	1607
<u>8.103.57 pg_lo_tell() : Retourne la position courante dans un objet de grande taille PostgreSQL.....</u>	1608
<u>8.103.58 pg_lo_unlink() : Efface un objet de grande taille PostgreSQL.....</u>	1608
<u>8.103.59 pg_lo_write() : Écrit un objet de grande taille PostgreSQL.....</u>	1608
<u>8.103.60 pg_meta_data() : Lit les métadonnées de la table PostgreSQL.....</u>	1608
<u>8.103.61 pg_num_fields() : Retourne le nombre de champ.....</u>	1608
<u>8.103.62 pg_num_rows() : Retourne le nombre de lignes PostgreSQL.....</u>	1609
<u>8.103.63 pg_options() : Retourne les options PostgreSQL.....</u>	1609

Sommaire

8 Référence des fonctions

8.103.64 <u>pg_parameter_status()</u> : Consulte un paramètre de configuration courant du serveur.....	1609
8.103.65 <u>pg_pconnect()</u> : Établit une connexion PostgreSQL persistante.....	1609
8.103.66 <u>pg_ping()</u> : Ping la connexion à la base.....	1610
8.103.67 <u>pg_port()</u> : Retourne le numéro de port.....	1610
8.103.68 <u>pg_prepare()</u> : Envoie une requête pour créer une requête préparée avec les paramètres donnés et attend l'exécution.....	1610
8.103.69 <u>pg_put_line()</u> : Envoie une chaîne au serveur PostgreSQL.....	1611
8.103.70 <u>pg_query_params()</u> : Envoie une commande au serveur et attend le résultat, avec les capacités de passer des paramètres séparément de la commande texte SQL.....	1611
8.103.71 <u>pg_query()</u> : Exécute une requête PostgreSQL.....	1611
8.103.72 <u>pg_result_error_field()</u> : Retourne un champ individuel d'un rapport d'erreur.....	1612
8.103.73 <u>pg_result_error()</u> : Lit le message d'erreur associé à un résultat.....	1612
8.103.74 <u>pg_result_seek()</u> : Modifie la ligne courant dans un résultat.....	1612
8.103.75 <u>pg_result_status()</u> : Lit le statut du résultat.....	1612
8.103.76 <u>pg_select()</u> : Effectue une sélection PostgreSQL.....	1613
8.103.77 <u>pg_send_execute()</u> : Envoie une requête pour exécuter une requête préparée avec des paramètres donnés, sans attendre le(s) résultat(s).....	1613
8.103.78 <u>pg_send_prepare()</u> : Envoie une requête pour créer une requête préparée avec les paramètres donnés, sans attendre la fin de son exécution.....	1613
8.103.79 <u>pg_send_query_params()</u> : Envoie une commande et sépare les paramètres au serveur sans attendre le(s) résultat(s).....	1614
8.103.80 <u>pg_send_query()</u> : Exécute une requête PostgreSQL asynchrone.....	1614
8.103.81 <u>pg_set_client_encoding()</u> : Choisit l'encodage du client PostgreSQL.....	1614
8.103.82 <u>pg_set_error_verbosity()</u> : Détermine la le degré des messages retournés par <u>pg_last_error</u> et <u>pg_result_error</u>	1614
8.103.83 <u>pg_trace()</u> : Active le suivi d'une connexion PostgreSQL.....	1615
8.103.84 <u>pg_transaction_status()</u> : Retourne le statut de la transaction en cours du serveur.....	1615
8.103.85 <u>pg_tty()</u> : Retourne le nom de TTY associé à la connexion.....	1615
8.103.86 <u>pg_unescape_bytea()</u> : Supprime le protection d'une chaîne de type <u>bytea</u>	1616
8.103.87 <u>pg_untrace()</u> : Termine le suivi d'une connexion PostgreSQL.....	1616
8.103.88 <u>pg_update()</u> : Modifie les lignes d'une table.....	1616
8.103.89 <u>pg_version()</u> : Retourne un tableau avec les versions du client, du protocole et du serveur (si disponible).....	1616
8.104 Fonctions POSIX.....	1617
8.104.1 Introduction.....	1617
8.104.2 Installation.....	1617
8.104.3 Constantes pré-définies.....	1617
8.104.4 Voir aussi.....	1618
8.104.6 <u>posix_ctermid()</u> : Retourne le chemin du terminal.....	1618
8.104.7 <u>posix_get_last_error()</u> : Lit le dernier numéro d'erreur généré par la dernière fonction POSIX qui a échoué.....	1619
8.104.8 <u>posix_getcwd()</u> : Chemin du dossier de travail courant.....	1619
8.104.9 <u>posix_getegid()</u> : Retourne l'ID effectif du groupe du processus courant.....	1619
8.104.10 <u>posix_geteuid()</u> : Retourne l'UID effectif de l'utilisateur du processus courant.....	1619
8.104.11 <u>posix_getgid()</u> : Retourne l'UID du groupe du processus courant.....	1619

Sommaire

8 Référence des fonctions

<u>8.104.12 posix_getgrgid()</u> : Retourne des informations sur un groupe.....	1619
<u>8.104.13 posix_getgrnam()</u> : Retourne des informations sur un groupe.....	1620
<u>8.104.14 posix_getgroups()</u> : Retourne les identifiants du groupe du processus courant.....	1620
<u>8.104.15 posix_getlogin()</u> : Retourne le nom de login.....	1621
<u>8.104.16 posix_getpgid()</u> : Retourne l'id du groupe de processus.....	1621
<u>8.104.17 posix_getpgrp()</u> : Retourne l'identifiant du groupe de processus.....	1621
<u>8.104.18 posix_getpid()</u> : Retourne l'identifiant du processus courant.....	1621
<u>8.104.19 posix_getppid()</u> : Retourne l'identifiant du processus parent.....	1621
<u>8.104.20 posix_getpwnam()</u> : Retourne des informations sur un utilisateur.....	1621
<u>8.104.21 posix_getpwuid()</u> : Retourne des informations sur un utilisateur.....	1622
<u>8.104.22 posix_getrlimit()</u> : Retourne les limites système.....	1622
<u>8.104.23 posix_getsid()</u> : Retourne le sid du processus.....	1623
<u>8.104.24 posix_getuid()</u> : Retourne l'ID de l'utilisateur du processus courant.....	1623
<u>8.104.25 posix_isatty()</u> : Détermine si un pointeur de fichier est un terminal interactif.....	1623
<u>8.104.26 posix_kill()</u> : Envoie un signal à un processus.....	1623
<u>8.104.27 posix_mkfifo()</u> : Crée un fichier FIFO (first in, first out) (un pipe nommé).....	1623
<u>8.104.28 posix_mknod()</u> : Crée un fichier spécial ou ordinaire (POSIX.1).....	1624
<u>8.104.29 posix_setegid()</u> : Modifie le GID réel du processus courant.....	1624
<u>8.104.30 posix seteuid()</u> : Modifie l'identifiant d'utilisateur réel (UID) du processus courant.....	1624
<u>8.104.31 posix_setgid()</u> : Fixe le GID effectif du processus courant.....	1624
<u>8.104.32 posix_setpgid()</u> : Fixe l'identifiant de groupe de processus.....	1624
<u>8.104.33 posix_setsid()</u> : Fait du processus courant un chef de session.....	1625
<u>8.104.34 posix_setuid()</u> : Fixe l'UID effective du processus courant.....	1625
<u>8.104.35 posix_strerror()</u> : Lit le message d'erreur associé à un numéro d'erreur POSIX.....	1625
<u>8.104.36 posix_times()</u> : Utilisation des ressources.....	1625
<u>8.104.37 posix_ttyname()</u> : Retourne le nom de device du terminal.....	1626
<u>8.104.38 posix_uname()</u> : Retourne le nom du système.....	1626
<u>8.105 Impression</u>	1627
<u>8.105.1 Introduction</u>	1627
<u>8.105.2 Installation</u>	1627
<u>8.105.3 Configuration à l'exécution</u>	1627
<u>8.105.5 printer_close()</u> : Ferme une connexion à une imprimante.....	1628
<u>8.105.6 printer_create_brush()</u> : Crée une nouvelle brosse.....	1628
<u>8.105.7 printer_create_dc()</u> : Crée un contexte d'imprimante.....	1628
<u>8.105.8 printer_create_font()</u> : Crée une nouvelle police.....	1629
<u>8.105.9 printer_create_pen()</u> : Crée un nouveau stylo.....	1630
<u>8.105.10 printer_delete_brush()</u> : Efface une brosse.....	1630
<u>8.105.11 printer_delete_dc()</u> : Efface un contexte d'imprimante.....	1630
<u>8.105.12 printer_delete_font()</u> : Efface une police.....	1630
<u>8.105.13 printer_delete_pen()</u> : Efface un stylo.....	1630
<u>8.105.14 printer_draw_bmp()</u> : Dessine un fichier BMP.....	1631
<u>8.105.15 printer_draw_chord()</u> : Dessine un arc de cercle.....	1631
<u>8.105.16 printer_draw_ellipse()</u> : Dessine une ellipse.....	1632
<u>8.105.17 printer_draw_line()</u> : Dessine une ligne.....	1632
<u>8.105.18 printer_draw_pie()</u> : Dessine un secteur angulaire.....	1633
<u>8.105.19 printer_draw_rectangle()</u> : Dessine un rectangle.....	1634
<u>8.105.20 printer_draw_roundrect()</u> : Dessine un rectangle, avec des coins	

Sommaire

8 Référence des fonctions

<u>arrondis</u>	1634
<u>8.105.21 printer_draw_text()</u> : Dessine un texte.....	1635
<u>8.105.22 printer_end_doc()</u> : Ferme le document.....	1635
<u>8.105.23 printer_end_page()</u> : Ferme la page active.....	1636
<u>8.105.24 printer_get_option()</u> : Lit les informations de configuration de l'imprimante.....	1636
<u>8.105.25 printer_list()</u> : Retourne la liste des imprimantes attachées à un serveur....	1636
<u>8.105.26 printer_logical_fontheight()</u> : Lit la taille logique de la police.....	1637
<u>8.105.27 printer_open()</u> : Ouvre une connexion à une imprimante.....	1637
<u>8.105.28 printer_select_brush()</u> : Sélectionne une brosse.....	1637
<u>8.105.29 printer_select_font()</u> : Sélectionne une police de caractères.....	1638
<u>8.105.30 printer_select_pen()</u> : Sélectionne un stylo.....	1638
<u>8.105.31 printer_set_option()</u> : Configure la connexion à l'imprimante.....	1639
<u>8.105.32 printer_start_doc()</u> : Commence un nouveau document.....	1640
<u>8.105.33 printer_start_page()</u> : Commence une nouvelle page.....	1641
<u>8.105.34 printer_write()</u> : Écrit des données sur l'imprimante.....	1641
<u>8.106 Création de document PostScript</u>	1642
<u>8.106.1 Introduction</u>	1642
<u>8.106.2 Pré-requis</u>	1642
<u>8.106.3 Installation</u>	1642
<u>8.106.4 Configuration à l'exécution</u>	1642
<u>8.106.5 Constantes pré-définies</u>	1642
<u>8.106.6 Contact Information</u>	1643
<u>8.106.8 ps_add_launchlink()</u> : Ajoute un lien qui démarre un fichier.....	1644
<u>8.106.9 ps_add_loclink()</u> : Ajoute un lien à une page dans le même document.....	1644
<u>8.106.10 ps_add_note()</u> : Ajoute une note à la page courante.....	1645
<u>8.106.11 ps_add_pdflink()</u> : Ajoute un lien à une page à un second document pdf.....	1645
<u>8.106.12 ps_add_weblink()</u> : Ajoute un lien à un emplacement web.....	1645
<u>8.106.13 ps_arc()</u> : Dessine un arc dans le sens contraire des aiguilles d'une montre.....	1645
<u>8.106.14 ps_arcn()</u> : Dessine un arc dans le sens des aiguilles d'une montre.....	1646
<u>8.106.15 ps_begin_page()</u> : Démarre une nouvelle page.....	1646
<u>8.106.16 ps_begin_pattern()</u> : Démarre un nouveau motif.....	1646
<u>8.106.17 ps_begin_template()</u> : Démarre un nouveau modèle.....	1646
<u>8.106.18 ps_circle()</u> : Dessine un cercle.....	1646
<u>8.106.19 ps_clip()</u> : Attache le dessin au chemin courant.....	1647
<u>8.106.20 ps_close_image()</u> : Ferme l'image et libère la mémoire.....	1647
<u>8.106.21 ps_close()</u> : Ferme le document PostScript.....	1647
<u>8.106.22 ps_closepath_stroke()</u> : Ferme et trace un chemin.....	1647
<u>8.106.23 ps_closepath()</u> : Ferme un chemin.....	1647
<u>8.106.24 ps_continue_text()</u> : Continue un texte à la ligne suivante.....	1647
<u>8.106.25 ps_curveto()</u> : Dessine une courbe.....	1648
<u>8.106.26 ps_delete()</u> : Supprime toutes les ressources d'un document PostScript.....	1648
<u>8.106.27 ps_end_page()</u> : Termine une page.....	1648
<u>8.106.28 ps_end_pattern()</u> : Termine un motif.....	1648
<u>8.106.29 ps_end_template()</u> : Termine un modèle.....	1648
<u>8.106.30 ps_fill_stroke()</u> : Remplit et trace le chemin courant.....	1648
<u>8.106.31 ps_fill()</u> : Remplit le chemin courant.....	1648
<u>8.106.32 ps_findfont()</u> : Charge une police.....	1649
<u>8.106.33 ps_get_buffer()</u> : Récupère le tampon complet contenant les données PS générées.....	1649

Sommaire

8 Référence des fonctions

<u>8.106.34 ps_get_parameter()</u> : Récupère certains paramètres.....	1649
<u>8.106.35 ps_get_value()</u> : Récupère certaines valeurs.....	1649
<u>8.106.36 ps_hyphenate()</u> : Relie un mot.....	1649
<u>8.106.37 ps_lineto()</u> : Dessine une ligne.....	1650
<u>8.106.38 ps_makespotcolor()</u> : Crée une tache de couleur.....	1650
<u>8.106.39 ps_moveto()</u> : Fixe le point courant.....	1650
<u>8.106.40 ps_new()</u> : Crée un nouvel objet document PostScript.....	1650
<u>8.106.41 ps_open_file()</u> : Ouvre un fichier pour écriture.....	1650
<u>8.106.42 ps_open_image_file()</u> : Ouvre une iamge à partir d'un fichier.....	1650
<u>8.106.43 ps_open_image()</u> : Lit une image pour placement futur.....	1651
<u>8.106.44 ps_place_image()</u> : Place une image sur la page.....	1651
<u>8.106.45 ps_rect()</u> : Dessine un rectangle.....	1651
<u>8.106.46 ps_restore()</u> : Restaure un contexte précédemment sauvegardé.....	1651
<u>8.106.47 ps_rotate()</u> : Fixe le facteur de rotation.....	1651
<u>8.106.48 ps_save()</u> : Sauvegarde le contexte courant.....	1651
<u>8.106.49 ps_scale()</u> : Fixe le facteur de redimensionnement.....	1652
<u>8.106.50 ps_set_border_color()</u> : Fixe la couleur des bordures pour les commentaires.....	1652
<u>8.106.51 ps_set_border_dash()</u> : Fixe la grandeur des tirets pour les bordures des commentaires.....	1652
<u>8.106.52 ps_set_border_style()</u> : Fixe le style de bordure des commentaires.....	1652
<u>8.106.53 ps_set_info()</u> : Fixe les champs d'informations d'un document.....	1652
<u>8.106.54 ps_set_parameter()</u> : Fixe certains paramètres.....	1652
<u>8.106.55 ps_set_text_pos()</u> : Fixe la position l'écriture du texte.....	1653
<u>8.106.56 ps_set_value()</u> : Fixe certaines valeurs.....	1653
<u>8.106.57 ps_setcolor()</u> : Fixe la couleur courante.....	1653
<u>8.106.58 ps_setdash()</u> : Fixe l'apparence d'une ligne pointillée.....	1653
<u>8.106.59 ps_setflat()</u> : Fixe la position à plat.....	1653
<u>8.106.60 ps_setfont()</u> : Fixe la police à utiliser pour la prochaine écriture.....	1653
<u>8.106.61 ps_setgray()</u> : Fixe la valeur de gris.....	1654
<u>8.106.62 ps_setlinecap()</u> : Fixe l'apparence des fins de ligne.....	1654
<u>8.106.63 ps_setlinejoin()</u> : Fixe comment les lignes connectés sont jointes.....	1654
<u>8.106.64 ps_setlinewidth()</u> : Fixe la largeur d'une ligne.....	1654
<u>8.106.65 ps_setmiterlimit()</u> : Fixe les limites de l'anglet.....	1654
<u>8.106.66 ps_setpolydash()</u> : Fixe l'apparence d'une ligne pointillée.....	1654
<u>8.106.67 ps_shading_pattern()</u> : Crée un motif basé sur le ton.....	1654
<u>8.106.68 ps_shading()</u> : Crée un ton pour usage futur.....	1655
<u>8.106.69 ps_shfill()</u> : Remplit un espace avec un ton.....	1655
<u>8.106.70 ps_show_boxed()</u> : Écriture de texte dans une boîte.....	1655
<u>8.106.71 ps_show_xy()</u> : Écrit du texte à la position donnée.....	1655
<u>8.106.72 ps_show()</u> : Écrit du texte.....	1655
<u>8.106.73 ps_string_geometry()</u> : Récupère les géométries d'une chaîne de caractères.....	1656
<u>8.106.74 ps_stringwidth()</u> : Récupère la largeur d'une chaîne de caractères.....	1656
<u>8.106.75 ps_stroke()</u> : Dessine le chemin courant.....	1656
<u>8.106.76 ps_symbol_name()</u> : Récupère le nom d'un glyphe.....	1656
<u>8.106.77 ps_symbol_width()</u> : Récupère la largeur d'un glyphe.....	1657
<u>8.106.78 ps_symbol()</u> : Écrit un glyphe.....	1657
<u>8.106.79 ps_translate()</u> : Fixe une translation.....	1657
<u>8.107 Pspell</u>	1658
<u>8.107.1 Introduction</u>	1658

Sommaire

8 Référence des fonctions

<u>8.107.2 Pré-requis</u>	1658
<u>8.107.3 Installation</u>	1658
<u>8.107.4 Configuration à l'exécution</u>	1658
<u>8.107.5 Types de ressources</u>	1658
<u>8.107.6 Constantes pré-définies</u>	1658
<u>8.107.8 pspell_add_to_session() : Ajoute le mot au dictionnaire personnel de la session courante</u>	1659
<u>8.107.9 pspell_check() : Vérifie un mot</u>	1659
<u>8.107.10 pspell_clear_session() : Remet à zéro la session courante</u>	1659
<u>8.107.11 pspell_config_create() : Crée une configuration utilisée pour ouvrir un dictionnaire</u>	1660
<u>8.107.12 pspell_config_data_dir() : Endroit où se trouvent les fichiers de données linguistiques</u>	1661
<u>8.107.13 pspell_config_dict_dir() : Endroit où se trouve le fichier global des mots</u>	1661
<u>8.107.14 pspell_config_ignore() : Ignore les mots de moins de N caractères</u>	1661
<u>8.107.15 pspell_config_mode() : Change le mode de suggestion</u>	1661
<u>8.107.16 pspell_config_personal() : Choisit le fichier qui contient le dictionnaire personnel</u>	1662
<u>8.107.17 pspell_config_repl() : Choisit le fichier qui contient les paires de remplacement</u>	1662
<u>8.107.18 pspell_config_runtogether() : Considère deux mots accolés comme un composé</u>	1662
<u>8.107.19 pspell_config_save_repl() : Active la sauvegarde des paires de remplacement</u>	1663
<u>8.107.20 pspell_new_config() : Charge un nouveau dictionnaire</u>	1663
<u>8.107.21 pspell_new_personal() : Charge un nouveau dictionnaire avec un dictionnaire personnel</u>	1663
<u>8.107.22 pspell_new() : Charge un nouveau dictionnaire</u>	1665
<u>8.107.23 pspell_save_wordlist() : Sauve le dictionnaire personnel dans un fichier</u>	1665
<u>8.107.24 pspell_store_replacement() : Enregistre une paire de remplacement pour un mot</u>	1666
<u>8.107.25 pspell_suggest() : Suggère une orthographe</u>	1666
<u>8.108 qtdom</u>	1667
<u>8.108.1 Pré-requis</u>	1667
<u>8.108.2 Installation</u>	1667
<u>8.108.4 qdom_tree() : Crée une arbre à partir d'une chaîne XML</u>	1667
<u>8.109 Radius</u>	1668
<u>8.109.1 Introduction</u>	1668
<u>8.109.2 Installation</u>	1668
<u>8.109.3 Constantes pré-définies</u>	1668
<u>8.109.4 Démarrage rapide</u>	1672
<u>8.109.5 Contact</u>	1673
<u>8.109.7 radius_add_server() : Ajoute un serveur</u>	1674
<u>8.109.8 radius_auth_open() : Crée une ressource Radius pour l'identification</u>	1674
<u>8.109.9 radius_close() : Libère toutes les ressources</u>	1674
<u>8.109.10 radius_config() : Demande à la bibliothèque de lire un fichier de configuration donné</u>	1674
<u>8.109.11 radius_create_request() : Crée une demande de compte ou d'identification</u>	1674
<u>8.109.12 radius_cvt_addr() : Convertit des données brutes en adresse IP</u>	1674
<u>8.109.13 radius_cvt_int() : Convertit des données brutes en entier</u>	1675

Sommaire

8 Référence des fonctions

<u>8.109.14 radius_cvt_string()</u> : Convertit des données brutes en chaîne de caractères.....	1675
<u>8.109.15 radius_demangle_mppe_key()</u> : Dérive les clés mppe depuis des données.....	1675
<u>8.109.16 radius_demangle()</u> : Assèche des données.....	1675
<u>8.109.17 radius_get_attr()</u> : Extrait un attribut.....	1675
<u>8.109.18 radius_get_vendor_attr()</u> : Extrait un attribut spécifique au vendeur.....	1675
<u>8.109.19 radius_put_addr()</u> : Attache une adresse IP en tant qu'attribut.....	1675
<u>8.109.20 radius_put_attr()</u> : Attache un attribut binaire.....	1676
<u>8.109.21 radius_put_int()</u> : Attache un attribut entier.....	1676
<u>8.109.22 radius_put_string()</u> : Attache un attribut chaîne de caractères.....	1676
<u>8.109.23 radius_put_vendor_addr()</u> : Attache un attribut IP-Address spécifique à un vendeur.....	1676
<u>8.109.24 radius_put_vendor_attr()</u> : Attache un attribut binaire à un vendeur spécifique.....	1676
<u>8.109.25 radius_put_vendor_int()</u> : Attache un attribut entier à un vendeur spécifique.....	1676
<u>8.109.26 radius_put_vendor_string()</u> : Attache un attribut sous la forme d'une chaîne à un vendeur spécifique.....	1677
<u>8.109.27 radius_request_authenticator()</u> : Retourne l'identificateur demandé.....	1677
<u>8.109.28 radius_send_request()</u> : Envoie la demande et attend une réponse.....	1677
<u>8.109.29 radius_server_secret()</u> : Retourne le secret partagé.....	1677
<u>8.109.30 radius_strerror()</u> : Retourne un message d'erreur.....	1677
<u>8.110 Rar</u>	1678
<u>8.110.1 Introduction</u>	1678
<u>8.110.2 Pré-requis</u>	1678
<u>8.110.3 Configuration à l'exécution</u>	1678
<u>8.110.4 Installation</u>	1678
<u>8.110.5 Types de ressources</u>	1678
<u>8.110.6 Constantes pré-définies</u>	1678
<u>8.110.7 Exemples</u>	1679
<u>8.110.9 rar_entry_get()</u> : Lit un élément dans l'archive Rar.....	1679
<u>8.110.10 Rar::extract()</u> : Extrait un élément d'une archive Rar.....	1680
<u>8.110.11 Rar::getAttr()</u> : Lit les attributs d'un élément d'archive Rar.....	1680
<u>8.110.12 Rar::getCrc()</u> : Lit la somme de contrôle d'un élément d'archive Rar.....	1681
<u>8.110.13 Rar::getFileTime()</u> : Lit la date de dernière modification d'un élément d'archive Rar.....	1682
<u>8.110.14 Rar::getHostOs()</u> : Lit l'OS d'un élément d'archive Rar.....	1682
<u>8.110.15 Rar::getMethod()</u> : Lit la méthode de compression d'un élément d'archive Rar.....	1683
<u>8.110.16 Rar::getName()</u> : Lit le nom de l'élément d'archive Rar.....	1683
<u>8.110.17 Rar::getPackedSize()</u> : Lit la taille compressée d'un élément d'archive Rar.....	1683
<u>8.110.18 Rar::getUnpackedSize()</u> : Lit la taille décompressée d'un élément d'archive Rar.....	1684
<u>8.110.19 Rar::getVersion()</u> : Lit la version utilisée pour créer le dernier élément Rar.....	1684
<u>8.110.20 rar_list()</u> : Liste les éléments de l'archive Rar.....	1685
<u>8.110.21 rar_open()</u> : Ouvre une archive Rar.....	1685
<u>8.111 Readline (GNU)</u>	1686
<u>8.111.1 Introduction</u>	1686

Sommaire

8 Référence des fonctions

<u>8.111.2</u>	<u>Pré-requis</u>	1686
<u>8.111.3</u>	<u>Installation</u>	1686
<u>8.111.4</u>	<u>Configuration à l'exécution</u>	1686
<u>8.111.5</u>	<u>Types de ressources</u>	1686
<u>8.111.6</u>	<u>Constantes pré-définies</u>	1686
<u>8.111.8</u>	<u>readline_callback_handler_install() : Initialise l'interface et le terminal de callback de readline, affiche le prompt et retourne immédiatement</u>	1687
<u>8.111.9</u>	<u>readline_callback_handler_remove() : Efface un gestionnaire de callback installé précédemment et restaure les paramètres du terminal</u>	1687
<u>8.111.10</u>	<u>readline_callback_read_char() : Lit un caractère et informe l'interface de callback readline lorsqu'une ligne est reçue</u>	1687
<u>8.111.11</u>	<u>readline_clear_history() : Efface l'historique</u>	1687
<u>8.111.12</u>	<u>readline_completion_function() : Enregistre une fonction de complétion</u>	1688
<u>8.111.13</u>	<u>readline_info() : Lit/modifie diverses variables internes</u>	1688
<u>8.111.14</u>	<u>readline_list_history() : Liste l'historique</u>	1688
<u>8.111.15</u>	<u>readline_on_new_line() : Informe readline que le curseur est passé à une nouvelle ligne</u>	1688
<u>8.111.16</u>	<u>readline_read_history() : Lit l'historique</u>	1688
<u>8.111.17</u>	<u>readline_redisplay() : Demande à readline de refaire l'affichage</u>	1689
<u>8.111.18</u>	<u>readline_write_history() : Ecrit dans l'historique</u>	1689
<u>8.111.19</u>	<u>readline() : Lit une ligne</u>	1689
<u>8.112</u>	<u>Recode (GNU)</u>	1690
<u>8.112.1</u>	<u>Introduction</u>	1690
<u>8.112.2</u>	<u>Pré-requis</u>	1690
<u>8.112.3</u>	<u>Installation</u>	1690
<u>8.112.4</u>	<u>Configuration à l'exécution</u>	1690
<u>8.112.5</u>	<u>Types de ressources</u>	1690
<u>8.112.6</u>	<u>Constantes pré-définies</u>	1691
<u>8.112.8</u>	<u>recode_string() : Recode une chaîne en fonction de la requête</u>	1691
<u>8.112.9</u>	<u>recode() : Alias de recode_string</u>	1691
<u>8.113</u>	<u>Expressions rationnelles</u>	1692
<u>8.113.1</u>	<u>Introduction</u>	1692
<u>8.113.2</u>	<u>Pré-requis</u>	1692
<u>8.113.3</u>	<u>Installation</u>	1692
<u>8.113.4</u>	<u>Configuration à l'exécution</u>	1692
<u>8.113.5</u>	<u>Types de ressources</u>	1692
<u>8.113.6</u>	<u>Constantes pré-définies</u>	1692
<u>8.113.7</u>	<u>Exemples</u>	1693
<u>8.113.8</u>	<u>Voir aussi</u>	1693
<u>8.113.10</u>	<u>ereg() : Expression rationnelle standard</u>	1694
<u>8.113.11</u>	<u>eregi_replace() : Remplacement par expression rationnelle insensible à la casse</u>	1694
<u>8.113.12</u>	<u>eregi() : Recherche par expression rationnelle insensible à la casse</u>	1695
<u>8.113.13</u>	<u>split() : Scinde une chaîne en un tableau, grâce à une expression rationnelle</u>	1695
<u>8.113.14</u>	<u>spliti() : Scinde une chaîne en un tableau, grâce à une expression rationnelle</u>	1696
<u>8.113.15</u>	<u>sql_regcase() : Prépare une expression rationnelle pour effectuer une recherche insensible à la casse</u>	1696
<u>8.114</u>	<u>Fonctions runkit</u>	1698
<u>8.114.1</u>	<u>Introduction</u>	1698

Sommaire

8 Référence des fonctions

<u>8.114.2 Pré-requis</u>	1698
<u>8.114.3 Configuration à l'exécution</u>	1698
<u>8.114.4 Constantes pré-définies</u>	1699
<u>8.114.6 Runkit_Sandbox_Parent() : Classe Anti-Sandbox Runkit</u>	1700
<u>8.114.7 runkit_class_adopt() : Convertit une classe de base à une classe héritée, ajoute une méthode ancestrale lorsque approprié</u>	1701
<u>8.114.8 runkit_class_emancipate() : Convertit une classe héritée à une classe de base, supprime toute méthode pour qui la portée est ancestrale</u>	1701
<u>8.114.9 runkit_constant_add() : Similaire à define(), mais permet aussi la définition dans définitions de classe</u>	1701
<u>8.114.10 runkit_constant_redefine() : Redéfinit une constante déjà définie</u>	1701
<u>8.114.11 runkit_constant_remove() : Enlève/Supprime une constante déjà définie</u>	1701
<u>8.114.12 runkit_function_add() : Ajoute une nouvelle fonction, similaire à create_function</u>	1701
<u>8.114.13 runkit_function_copy() : Copie une fonction vers un nom de fonction nouveau</u>	1701
<u>8.114.14 runkit_function_redefine() : Remplace une définition de fonction avec une nouvelle implémentation</u>	1702
<u>8.114.15 runkit_function_remove() : Enlève une définition de fonction</u>	1702
<u>8.114.16 runkit_function_rename() : Change le nom d'une fonction</u>	1702
<u>8.114.17 runkit_import() : Traite un fichier PHP important fonctions et définitions de classes, écrasement où applicable</u>	1702
<u>8.114.18 runkit_lint_file() : Vérifie la syntaxe PHP d'un fichier spécifié</u>	1702
<u>8.114.19 runkit_lint() : Vérifie la syntaxe PHP de code PHP spécifié</u>	1703
<u>8.114.20 runkit_method_add() : Ajoute dynamiquement une nouvelle méthode à une classe donnée</u>	1703
<u>8.114.21 runkit_method_copy() : Copie une méthode d'une classe à une autre</u>	1703
<u>8.114.22 runkit_method_redefine() : Change dynamiquement le code de la méthode donnée</u>	1703
<u>8.114.23 runkit_method_remove() : Supprime dynamiquement la méthode donnée</u>	1704
<u>8.114.24 runkit_method_rename() : Change dynamiquement le nom de la méthode donnée</u>	1704
<u>8.114.25 runkit_return_value_used() : Détermine si la valeur de retour des fonctions courantes sera utilisée</u>	1704
<u>8.114.26 runkit_sandbox_output_handler() : Spécifie une fonction à capturer et/ou traiter la sortie à partir d'un runkit sandbox</u>	1704
<u>8.114.27 runkit_superglobals() : Retourne un tableau indexé numériquement des variables superglobales enregistrées</u>	1705
<u>8.115 Sémaphores et gestion de la mémoire partagée</u>	1706
<u>8.115.1 Introduction</u>	1706
<u>8.115.2 Pré-requis</u>	1706
<u>8.115.3 Installation</u>	1706
<u>8.115.4 Configuration à l'exécution</u>	1706
<u>8.115.5 Types de ressources</u>	1707
<u>8.115.6 Constantes pré-définies</u>	1707
<u>8.115.8 msg_get_queue() : Crée ou s'attache à une file de messages</u>	1707
<u>8.115.9 msg_receive() : Reçoit un message depuis une file de messages</u>	1708
<u>8.115.10 msg_remove_queue() : Détruit une file de messages</u>	1708
<u>8.115.11 msg_send() : Envoie un message dans une file</u>	1709

Sommaire

8 Référence des fonctions

<u>8.115.12 msg_set_queue()</u> : Modifie des informations dans la file de messages.....	1709
<u>8.115.13 msg_stat_queue()</u> : Retourne des informations sur la file de messages....	1709
<u>8.115.14 sem_acquire()</u> : Réserve un sémaphore.....	1710
<u>8.115.15 sem_get()</u> : Retourne un identifiant de sémaphore.....	1710
<u>8.115.16 sem_release()</u> : Libère un sémaphore.....	1711
<u>8.115.17 sem_remove()</u> : Détruit un sémaphore.....	1711
<u>8.115.18 shm_attach()</u> : Crée ou ouvre un segment de mémoire partagée.....	1711
<u>8.115.19 shm_detach()</u> : Libère un segment de mémoire partagée.....	1712
<u>8.115.20 shm_get_var()</u> : Lit une variable dans la mémoire partagée.....	1712
<u>8.115.21 shm_put_var()</u> : Insère ou modifie une variable de la mémoire partagée..	1712
<u>8.115.22 shm_remove_var()</u> : Efface une variable de la mémoire partagée.....	1712
<u>8.115.23 shm_remove()</u> : Supprime un segment de mémoire partagée sous Unix..	1712
8.116 SESAM.....	1714
<u>8.116.1 Introduction.....</u>	1714
<u>8.116.2 Configuration à l'exécution.....</u>	1714
<u>8.116.3 Notes sur la configuration.....</u>	1715
<u>8.116.4 Considérations à l'exécution.....</u>	1716
<u>8.116.5 Types de curseurs.....</u>	1716
<u>8.116.6 Portage.....</u>	1717
<u>8.116.7 Sécurité.....</u>	1717
<u>8.116.8 Migration d'une autre base SQL.....</u>	1717
<u>8.116.9 Notes sur l'utilisation de types SQL divers.....</u>	1718
<u>8.116.10 Support des "champs multiples" de SESAM.....</u>	1718
<u>8.116.11 Voir aussi.....</u>	1719
<u>8.116.13 sesam_commit()</u> : Valide la transaction SESAM en cours.....	1719
<u>8.116.14 sesam_connect()</u> : Ouvre une connexion SESAM.....	1720
<u>8.116.15 sesam_diagnostic()</u> : Retourne l'état de la dernière requête SESAM.....	1720
<u>8.116.16 sesam_disconnect()</u> : Déconnexion d'une base SESAM.....	1722
<u>8.116.17 sesam_errormsg()</u> : Retourne le message d'erreur SESAM.....	1722
<u>8.116.18 sesam_execimm()</u> : Exécute immédiatement une requête SQL.....	1723
<u>8.116.19 sesam_fetch_array()</u> : Lit une ligne dans un tableau associatif.....	1723
<u>8.116.20 sesam_fetch_result()</u> : Retourne tout ou partie d'un résultat SESAM.....	1724
<u>8.116.21 sesam_fetch_row()</u> : Lit une ligne dans un tableau.....	1725
<u>8.116.22 sesam_field_array()</u> : Retourne des informations sur une colonne de <u> résultat.....</u>	1726
<u>8.116.23 sesam_field_name()</u> : Retourne le nom d'une colonne.....	1728
<u>8.116.24 sesam_free_result()</u> : Libère les ressources SESAM.....	1728
<u>8.116.25 sesam_num_fields()</u> : Retourne le nombre de colonnes dans un résultat..	1728
<u>8.116.26 sesam_query()</u> : Exécute une requête SESAM.....	1729
<u>8.116.27 sesam_rollback()</u> : Annule une transaction SESAM.....	1730
<u>8.116.28 sesam_seek_row()</u> : Déplace un curseur à défilement.....	1730
<u>8.116.29 sesam_settransaction()</u> : Modifie les paramètres de transaction SESAM..	1731
8.117 Sessions.....	1733
<u>8.117.1 Introduction.....</u>	1733
<u>8.117.2 Sessions et sécurité.....</u>	1733
<u>8.117.3 Pré-requis.....</u>	1734
<u>8.117.4 Installation.....</u>	1734
<u>8.117.5 Configuration à l'exécution.....</u>	1734
<u>8.117.6 Types de ressources.....</u>	1739
<u>8.117.7 Constantes pré-définies.....</u>	1739
<u>8.117.8 Exemples.....</u>	1739

Sommaire

8 Référence des fonctions

<u>8.117.9 Passer l'identifiant de session (session ID)</u>	1740
<u>8.117.10 Gestion personnalisée des sessions</u>	1741
<u>8.117.12 session_cache_limiter() : Lit et/ou modifie le limiteur de cache de session</u>	1742
<u>8.117.13 session_commit() : Alias de session_write_close</u>	1743
<u>8.117.14 session_decode() : Décode les données de session</u>	1743
<u>8.117.15 session_destroy() : Détruit une session</u>	1743
<u>8.117.16 session_encode() : Encode les données de session</u>	1744
<u>8.117.17 session_get_cookie_params() : Lit la configuration du cookie de session</u>	1744
<u>8.117.18 session_id() : Lit et/ou modifie l'identifiant courant de session</u>	1744
<u>8.117.19 session_is_registered() : Vérifie si une variable est enregistrée dans la session</u>	1744
<u>8.117.20 session_module_name() : Lit et/ou modifie le module de session courant</u>	1745
<u>8.117.21 session_name() : Lit et/ou modifie le nom de la session</u>	1745
<u>8.117.22 session_regenerate_id() : Remplace l'identifiant de session courant par un nouveau</u>	1745
<u>8.117.23 session_register() : Enregistre une variable dans une session</u>	1745
<u>8.117.24 session_save_path() : Lit et/ou modifie le chemin de sauvegarde des sessions</u>	1747
<u>8.117.25 session_set_cookie_params() : Modifie les paramètres du cookie de session</u>	1747
<u>8.117.26 session_set_save_handler() : Configure les fonctions de stockage de sessions</u>	1747
<u>8.117.27 session_start() : Initialise une session</u>	1749
<u>8.117.28 session_unregister() : Supprime une variable de la session</u>	1750
<u>8.117.29 session_unset() : Détruit toutes les variables d'une session</u>	1750
<u>8.117.30 session_write_close() : Ecrit les données de session et ferme la session</u> ..	1751
<u>8.118 Mémoire partagée</u>	1752
<u>8.118.1 Introduction</u>	1752
<u>8.118.2 Pré-requis</u>	1752
<u>8.118.3 Installation</u>	1752
<u>8.118.4 Configuration à l'exécution</u>	1752
<u>8.118.5 Types de ressources</u>	1752
<u>8.118.6 Constantes pré-définies</u>	1752
<u>8.118.7 Exemples</u>	1752
<u>8.118.9 shmop_delete() : Détruit un bloc de mémoire partagée</u>	1753
<u>8.118.10 shmop_open() : Crée ou ouvre un bloc de mémoire partagée</u>	1754
<u>8.118.11 shmop_read() : Lit des données à partir d'un bloc</u>	1754
<u>8.118.12 shmop_size() : Lire la taille du bloc de mémoire partagée</u>	1755
<u>8.118.13 shmop_write() : Ecrire dans un bloc de mémoire partagée</u>	1755
<u>8.119 Fonctions SimpleXML</u>	1756
<u>8.119.1 Introduction</u>	1756
<u>8.119.2 Pré-requis</u>	1756
<u>8.119.3 Installation</u>	1756
<u>8.119.4 Exemples</u>	1756
<u>8.119.6 SimpleXMLElement->attributes() : Identifie les attributs d'un élément</u>	1759
<u>8.119.7 SimpleXMLElement->children() : Trouve les enfants d'une node</u>	1759
<u>8.119.8 SimpleXMLElement->xpath() : Exécute une requête Xpath sur des données XML</u>	1760

Sommaire

8 Référence des fonctions

8.119.9 <u>simplexml_import_dom()</u> : Construit un objet SimpleXMLElement à partir d'une node DOM.....	1761
8.119.10 <u>simplexml_load_file()</u> : Convertit un fichier XML en objet.....	1761
8.119.11 <u>simplexml_load_string()</u> : Convertit une chaîne XML en objet.....	1762
8.120 Fonctions SNMP	1764
8.120.1 <u>Introduction</u>	1764
8.120.2 <u>Pré-requis</u>	1764
8.120.3 <u>Installation</u>	1764
8.120.4 <u>Configuration à l'exécution</u>	1764
8.120.5 <u>Types de ressources</u>	1764
8.120.6 <u>Constantes pré-définies</u>	1764
8.120.8 <u>snmp_get_valueretrieval()</u> : Retourne la méthode avec laquelle les valeurs SNMP seront retournées.....	1765
8.120.9 <u>snmp_read_mib()</u> : Lit et analyse un fichier MIB dans l'arbre actif MIB.....	1765
8.120.10 <u>snmp_set_enum_print()</u> : Retourne toutes les valeurs qui sont des énumérations avec leur valeur d'énumération au lieu de l'entier.....	1765
8.120.11 <u>snmp_set_oid_numeric_print()</u> : Retourne tous les objets y compris leur identifiant d'objet dans celui spécifié.....	1766
8.120.12 <u>snmp_set_quick_print()</u> : Ecrit la valeur courante de l'option quick_print de la bibliothèque UCD.....	1766
8.120.13 <u>snmp_set_valueretrieval()</u> : Spécifie la méthode avec laquelle les valeurs SNMP seront retournées.....	1766
8.120.14 <u>snmpget()</u> : Reçoit un objet SNMP.....	1767
8.120.15 <u>snmpgetnext()</u> : Retourne un objet SNMP.....	1767
8.120.16 <u>snmprealwalk()</u> : Retourne tous les objets, y compris leur ID d'objet.....	1767
8.120.17 <u>snmpset()</u> : Configure un objet SNMP.....	1767
8.120.18 <u>snmpwalk()</u> : Reçoit tous les objets SNMP d'un agent.....	1768
8.120.19 <u>snmpwalkoid()</u> : Demande d'informations d'arbre sur une entité du réseau.....	1768
8.121 Fonctions SOAP	1770
8.121.1 <u>Introduction</u>	1770
8.121.2 <u>Pré-requis</u>	1770
8.121.3 <u>Installation</u>	1770
8.121.4 <u>Configuration à l'exécution</u>	1770
8.121.5 <u>Classes pré-définies</u>	1770
8.121.6 <u>Constantes pré-définies</u>	1772
8.121.8 <u>SoapClient-> __call()</u> : Appelle une fonction SOAP (dépréciée).....	1774
8.121.9 <u>SoapClient-> __construct()</u> : Constructeur SoapClient.....	1774
8.121.10 <u>SoapClient-> doRequest()</u> : Effectue une requête SOAP.....	1774
8.121.11 <u>SoapClient-> getFunctions()</u> : Retourne une liste de fonctions SOAP.....	1775
8.121.12 <u>SoapClient-> __getLastRequest()</u> : Retourne la dernière requête SOAP.....	1775
8.121.13 <u>SoapClient-> __getLastRequestHeaders()</u> : Retourne les en-têtes de la dernière requête SOAP.....	1775
8.121.14 <u>SoapClient-> __getLastResponse()</u> : Retourne la dernière réponse SOAP.....	1775
8.121.15 <u>SoapClient-> __getLastResponseHeaders()</u> : Retourne les en-têtes de la dernière réponse SOAP.....	1775
8.121.16 <u>SoapClient-> __getTypes()</u> : Retourne une liste des types SOAP.....	1775
8.121.17 <u>SoapClient-> __setCookie()</u> : Définit le cookie qui sera envoyé avec la	

Sommaire

8 Référence des fonctions

<u>requête SOAP</u>	1776
<u>8.121.18 SoapClient-> soapCall()() : Appel une fonction SOAP</u>	1776
<u>8.121.19 SoapFault-> construct()() : Constructeur SoapFault</u>	1776
<u>8.121.20 SoapHeader-> construct()() : Constructeur SoapHeader</u>	1776
<u>8.121.21 SoapParam-> construct()() : Constructeur SoapParam</u>	1776
<u>8.121.22 SoapServer->addFunction()() : Ajoute une ou plusieurs fonctions qui vont gérer les requêtes SOAP</u>	1777
<u>8.121.23 SoapServer-> construct()() : Constructeur SoapServer</u>	1777
<u>8.121.24 SoapServer->fault()() : Provoque une erreur SoapServer indiquant une erreur</u>	1777
<u>8.121.25 SoapServer->getFunctions()() : Retourne la liste des fonctions définies</u> ...	1777
<u>8.121.26 SoapServer->handle()() : Gère une requête SOAP</u>	1777
<u>8.121.27 SoapServer->setClass()() : Définit la classe qui gère les requêtes SOAP</u>	1777
<u>8.121.28 SoapServer->setPersistence()() : Définit le mode persistant de SoapServer</u>	1778
<u>8.121.29 SoapVar-> construct()() : Constructeur SoapVar</u>	1778
<u>8.121.30 use_soap_error_handler() : Utilisation ou non du gestionnaire d'erreurs SOAP et retourne l'ancienne valeur</u>	1778
<u>8.122 Sockets</u>	1779
<u>8.122.1 Introduction</u>	1779
<u>8.122.2 Pré-requis</u>	1779
<u>8.122.3 Installation</u>	1779
<u>8.122.4 Configuration à l'exécution</u>	1779
<u>8.122.5 Types de ressources</u>	1779
<u>8.122.6 Constantes pré-définies</u>	1779
<u>8.122.7 Erreurs de socket</u>	1780
<u>8.122.8 Exemples</u>	1781
<u>8.122.10 socket_bind() : Lie un nom à une socket</u>	1783
<u>8.122.11 socket_clear_error() : Efface toutes les erreurs précédemment générées par une socket</u>	1783
<u>8.122.12 socket_close() : Ferme une socket</u>	1784
<u>8.122.13 socket_connect() : Crée une connexion sur une socket</u>	1784
<u>8.122.14 socket_create_listen() : Ouvre une socket sur un port pour accepter les connexions</u>	1784
<u>8.122.15 socket_create_pair() : Crée une paire de sockets identiques et les stocke dans un tableau</u>	1785
<u>8.122.16 socket_create() : Crée une socket</u>	1787
<u>8.122.17 socket_get_option() : Lit les options de la socket</u>	1788
<u>8.122.18 socket_getpeername() : Interroge l'autre extrémité de la communication</u> ...	1789
<u>8.122.19 socket_getsockname() : Interroge la socket locale</u>	1789
<u>8.122.20 socket_last_error() : Lit la dernière erreur générée par une socket</u>	1790
<u>8.122.21 socket_listen() : Attend une connexion sur une socket</u>	1790
<u>8.122.22 socket_read() : Lit des données d'une socket</u>	1791
<u>8.122.23 socket_recv() : Reçoit des données d'une socket connectée</u>	1791
<u>8.122.24 socket_recvfrom() : Reçoit des données d'une socket, connectée ou pas</u>	1791
<u>8.122.25 socket_select() : Exécute l'appel système select() un tableau de sockets avec une durée d'expiration</u>	1792
<u>8.122.26 socket_send() : Envoie des données à une socket connectée</u>	1793
<u>8.122.27 socket_sendto() : Envoie un message à une socket, qu'elle soit</u>	

Sommaire

8 Référence des fonctions

<u>connectée ou pas</u>	1794
<u>8.122.28 socket_set_block()</u> : Met la socket en mode bloquant.....	1794
<u>8.122.29 socket_set_nonblock()</u> : Sélectionne le mode non bloquant d'un descripteur de fichier.....	1795
<u>8.122.30 socket_set_option()</u> : Modifie les options de socket.....	1795
<u>8.122.31 socket_shutdown()</u> : Eteint une socket en lecture et/ou écriture.....	1796
<u>8.122.32 socket_strerror()</u> : Retourne une chaîne décrivant un message d'erreur....	1796
<u>8.122.33 socket_write()</u> : Écrit dans une socket.....	1796
<u>8.123 Fonctions Standard PHP Library (SPL)</u>	1798
<u>8.123.1 Introduction</u>	1798
<u>8.123.2 Installation</u>	1798
<u>8.123.3 Constantes pré-définies</u>	1798
<u>8.123.5 ArrayIterator::key()</u> : Retourne la clé courante du tableau.....	1800
<u>8.123.6 ArrayIterator::next()</u> : Se déplace vers la prochaine entrée.....	1800
<u>8.123.7 ArrayIterator::rewind()</u> : Revient à la position initiale.....	1801
<u>8.123.8 ArrayIterator::seek()</u> : Avance à une position.....	1801
<u>8.123.9 ArrayIterator::valid()</u> : Vérifie si un tableau contient d'autres entrées.....	1801
<u>8.123.10 ArrayObject::append()</u> : Ajoute la valeur à la fin.....	1801
<u>8.123.11 ArrayObject::__construct()</u> : Construit un nouvel objet tableau.....	1802
<u>8.123.12 ArrayObject::count()</u> : Retourne le nombre d'éléments dans l'itérateur.....	1802
<u>8.123.13 ArrayObject::getIterator()</u> : Créé un nouvel itérateur à partir d'une instance ArrayObject.....	1802
<u>8.123.14 ArrayObject::offsetExists()</u> : Retourne si \$index existe.....	1803
<u>8.123.15 ArrayObject::offsetGet()</u> : Retourne la valeur à l'\$index spécifié.....	1803
<u>8.123.16 ArrayObject::offsetSet()</u> : Définit \$newval comme valeur à l'\$index spécifié.....	1803
<u>8.123.17 ArrayObject::offsetUnset()</u> : Efface la valeur à l'\$index spécifié.....	1803
<u>8.123.18 CachingIterator::hasNext()</u> : Vérifie si l'itérateur interne a un élément suivant valide.....	1803
<u>8.123.19 CachingIterator::next()</u> : Déplace l'itérateur à la position suivante.....	1804
<u>8.123.20 CachingIterator::rewind()</u> : Remet l'itérateur au début.....	1804
<u>8.123.21 CachingIterator::toString()</u> : Retourne la représentation de l'élément courant sous forme de chaîne.....	1804
<u>8.123.22 CachingIterator::valid()</u> : Vérifie si l'élément courant est valide.....	1804
<u>8.123.23 CachingRecursiveIterator::getChildren()</u> : Retourne le fils de l'itérateur interne comme un CachingRecursiveIterator.....	1804
<u>8.123.24 CachingRecursiveIterator::hasChildren()</u> : Vérifie si l'élément courant de l'itérateur interne a un fils.....	1804
<u>8.123.25 class_implements()</u> : Retourne les interfaces implémentées dans une classe donnée.....	1805
<u>8.123.26 class_parents()</u> : Retourne la classe parente d'une classe.....	1805
<u>8.123.27 DirectoryIterator::__construct()</u> : Construit un nouvel itérateur de dossier à partir d'un chemin.....	1805
<u>8.123.28 DirectoryIterator::current()</u> : Retourne this (requis pour l'interface Iterator).....	1805
<u>8.123.29 DirectoryIterator::getATime()</u> : Récupère la date et l'heure du dernier accès à un fichier.....	1805
<u>8.123.30 DirectoryIterator::getChildren()</u> : Retourne un itérateur pour l'entrée courante si c'est un répertoire.....	1805
<u>8.123.31 DirectoryIterator::getCTime()</u> : Récupère l'heure de modification de l'inode d'un fichier.....	1806

Sommaire

8 Référence des fonctions

<u>8.123.32 DirectoryIterator::getFilename() : Retourne le nom de l'entrée courante du dossier.....</u>	1806
<u>8.123.33 DirectoryIterator::getGroup() : Récupère le groupe d'un fichier.....</u>	1806
<u>8.123.34 DirectoryIterator::getNode() : Récupère l'inode d'un fichier.....</u>	1806
<u>8.123.35 DirectoryIterator::getMTime() : Récupère l'heure de la dernière modification d'un fichier.....</u>	1806
<u>8.123.36 DirectoryIterator::getOwner() : Récupère le propriétaire d'un fichier.....</u>	1806
<u>8.123.37 DirectoryIterator::getPath() : Retourne le chemin du dossier.....</u>	1807
<u>8.123.38 DirectoryIterator::getPathname() : Retourne le chemin et le nom de l'entrée courante du dossier.....</u>	1807
<u>8.123.39 DirectoryIterator::getPerms() : Récupère les permissions d'un fichier.....</u>	1807
<u>8.123.40 DirectoryIterator::getSize() : Récupère la taille d'un fichier.....</u>	1807
<u>8.123.41 DirectoryIterator::getType() : Récupère le type d'un fichier.....</u>	1807
<u>8.123.42 DirectoryIterator::isDir() : Retourne TRUE si le fichier est un dossier.....</u>	1807
<u>8.123.43 DirectoryIterator::isDot() : Retourne TRUE si l'entrée courante est '.' ou '..'.....</u>	1807
<u>8.123.44 DirectoryIterator::isExecutable() : Retourne TRUE si le fichier est exécutable.....</u>	1808
<u>8.123.45 DirectoryIterator::isFile() : Retourne TRUE si l'entrée est un fichier valide.....</u>	1808
<u>8.123.46 DirectoryIterator::isLink() : Retourne TRUE si le fichier est un lien symbolique.....</u>	1808
<u>8.123.47 DirectoryIterator::isReadable() : Retourne TRUE si le fichier est accessible en lecture.....</u>	1808
<u>8.123.48 DirectoryIterator::isWritable() : Retourne TRUE si le fichier peut être modifié.....</u>	1808
<u>8.123.49 DirectoryIterator::key() : Retourne l'entrée courante du dossier.....</u>	1808
<u>8.123.50 DirectoryIterator::next() : Se déplace vers la prochaine entrée.....</u>	1809
<u>8.123.51 DirectoryIterator::rewind() : Revient au début du dossier.....</u>	1809
<u>8.123.52 DirectoryIterator::valid() : Vérifie si le répertoire contient encore des entrées.....</u>	1809
<u>8.123.53 FilterIterator::current() : Récupère la valeur de l'élément courant.....</u>	1809
<u>8.123.54 FilterIterator::getInnerIterator() : Récupère l'itérateur interne.....</u>	1809
<u>8.123.55 FilterIterator::key() : Récupère la clé courante.....</u>	1809
<u>8.123.56 FilterIterator::next() : Déplace l'itérateur à la position suivante.....</u>	1810
<u>8.123.57 FilterIterator::rewind() : Remplace l'itérateur au début.....</u>	1810
<u>8.123.58 FilterIterator::valid() : Vérifie si l'élément courant est valide.....</u>	1810
<u>8.123.59 iterator_count() : Compte de nombre d'éléments dans un itérateur.....</u>	1810
<u>8.123.60 iterator_to_array() : Copie un itérateur dans un tableau.....</u>	1810
<u>8.123.61 LimitIterator::getPosition() : Retourne la position courante.....</u>	1810
<u>8.123.62 LimitIterator::next() : Déplace l'itérateur à la position suivante.....</u>	1810
<u>8.123.63 LimitIterator::rewind() : Remplace l'itérateur au début.....</u>	1811
<u>8.123.64 LimitIterator::seek() : Place l'itérateur à une position donnée.....</u>	1811
<u>8.123.65 LimitIterator::valid() : Vérifie si l'élément courant est valide.....</u>	1811
<u>8.123.66 ParentIterator::getChildren() : Retourne le fils de l'itérateur interne contenu dans ParentIterator.....</u>	1811
<u>8.123.67 ParentIterator::hasChildren() : Vérifie si l'élément courant de l'itérateur interne a un fils.....</u>	1811
<u>8.123.68 ParentIterator::next() : Déplace l'itérateur à la position suivante.....</u>	1811
<u>8.123.69 ParentIterator::rewind() : Remplace l'itérateur au début.....</u>	1811
<u>8.123.70 RecursiveDirectoryIterator::getChildren() : Retourne un itérateur pour</u>	

Sommaire

8 Référence des fonctions

<u>l'entrée courante si c'est un dossier</u>	1812
<u>8.123.71 RecursiveDirectoryIterator::hasChildren() : Vérifie si l'entrée courante est un dossier et n'est pas '.' ou '..'</u>	1812
<u>8.123.72 RecursiveDirectoryIterator::key() : Retourne le chemin et le nom de l'entrée courante du dossier</u>	1812
<u>8.123.73 RecursiveDirectoryIterator::next() : Se déplace vers la prochaine entrée</u> ..	1812
<u>8.123.74 RecursiveDirectoryIterator::rewind() : Revient à la position initiale dans le dossier</u>	1812
<u>8.123.75 RecursiveIteratorIterator::current() : Accède à la valeur de l'élément courant</u>	1812
<u>8.123.76 RecursiveIteratorIterator::getDepth() : Récupère la profondeur courante de la récursivité de l'itérateur</u>	1813
<u>8.123.77 RecursiveIteratorIterator::getSubIterator() : L'itérateur secondaire actif courant</u>	1813
<u>8.123.78 RecursiveIteratorIterator::key() : Accède à la clé courante</u>	1813
<u>8.123.79 RecursiveIteratorIterator::next() : Déplace l'itérateur à la position suivante</u>	1813
<u>8.123.80 RecursiveIteratorIterator::rewind() : Remplace l'itérateur au début</u>	1813
<u>8.123.81 RecursiveIteratorIterator::valid() : Vérifie si la position courante est valide</u>	1813
<u>8.123.82 SimpleXMLIterator::current() : Retourne l'entrée courante de SimpleXML</u>	1814
<u>8.123.83 SimpleXMLIterator::getChildren() : Retourne un itérateur pour l'entrée courante, si c'est un objet SimpleXML</u>	1814
<u>8.123.84 SimpleXMLIterator::hasChildren() : Indique si l'entrée courante de SimpleXML est un objet</u>	1814
<u>8.123.85 SimpleXMLIterator::key() : Retourne la clé courante SimpleXML</u>	1814
<u>8.123.86 SimpleXMLIterator::next() : Va à l'entrée SimpleXML suivante</u>	1814
<u>8.123.87 SimpleXMLIterator::rewind() : Remplace le pointeur SimpleXML au début</u> ..	1814
<u>8.123.88 SimpleXMLIterator::valid() : Vérifie si une ressource SimpleXML contient d'autres entrées</u>	1815
<u>8.123.89 spl_classes() : Retourne les classes SPL disponibles</u>	1815
<u>8.124 Fonctions SQLite</u>	1816
<u>8.124.1 Introduction</u>	1816
<u>8.124.2 Installation</u>	1816
<u>8.124.3 Pré-requis</u>	1817
<u>8.124.4 Types de ressources</u>	1817
<u>8.124.5 Constantes pré-définies</u>	1817
<u>8.124.6 Classes pré-définies</u>	1818
<u>8.124.7 Configuration à l'exécution</u>	1820
<u>8.124.9 sqlite_busy_timeout() : SQLiteDatabase->busyTimeout()</u>	1821
<u>8.124.10 sqlite_changes() : SQLiteDatabase->changes()</u>	1821
<u>8.124.11 sqlite_close() : Ferme la connexion à SQLite</u>	1822
<u>8.124.12 sqlite_column() : SQLiteResult->column()</u>	1822
<u>8.124.13 sqlite_create_aggregate() : SQLiteDatabase->createAggregate()</u>	1822
<u>8.124.14 sqlite_create_function() : SQLiteDatabase->createFunction()</u>	1822
<u>8.124.15 sqlite_current() : SQLiteResult->current()</u>	1823
<u>8.124.16 sqlite_error_string() : Retourne le message d'erreur SQLite</u>	1823
<u>8.124.17 sqlite_escape_string() : Protège une chaîne de caractères pour utilisation avec SQLite</u>	1823
<u>8.124.18 sqlite_exec() : SQLiteDatabase->exec()</u>	1824

Sommaire

8 Référence des fonctions

<u>8.124.19 sqlite_factory()</u> : Ouvre une base SQLite et crée un objet pour elle.....	1824
<u>8.124.20 sqlite_fetch_all()</u> : SQLiteResult->fetchAll().	1824
<u>8.124.21 sqlite_fetch_array()</u> : SQLiteResult->fetch().	1824
<u>8.124.22 sqlite_fetch_column_types()</u> : SQLiteDatabase->fetchColumnTypes().	1825
<u>8.124.23 sqlite_fetch_object()</u> : SQLiteResult->fetchObject().	1825
<u>8.124.24 sqlite_fetch_single()</u> : SQLiteResult->fetchSingle().	1825
<u>8.124.25 sqlite_fetch_string()</u> : Alias de sqlite_fetch_single.	1825
<u>8.124.26 sqlite_field_name()</u> : SQLiteResult->fieldName().	1825
<u>8.124.27 sqlite_has_more()</u> : Indique s'il reste des lignes SQLite à lire.	1826
<u>8.124.28 sqlite_has_prev()</u> : SQLiteResult->hasPrev().	1826
<u>8.124.29 sqlite_key()</u> : SQLiteResult->key().	1826
<u>8.124.30 sqlite_last_error()</u> : SQLiteDatabase->lastError().	1826
<u>8.124.31 sqlite_last_insert_rowid()</u> : SQLiteDatabase->lastInsertRowid().	1827
<u>8.124.32 sqlite_libencoding()</u> : Retourne l'encodage utilisé par la bibliothèque SQLite.	1827
<u>8.124.33 sqlite_libversion()</u> : Retourne la version de la bibliothèque SQLite.	1827
<u>8.124.34 sqlite_next()</u> : SQLiteResult->next().	1827
<u>8.124.35 sqlite_num_fields()</u> : SQLiteResult->numFields().	1828
<u>8.124.36 sqlite_num_rows()</u> : SQLiteResult->numRows().	1828
<u>8.124.37 sqlite_open()</u> : Ouvre une base SQLite et la crée si elle n'existe pas.	1828
<u>8.124.38 sqlite_popen()</u> : Ouvre une connexion SQLite persistante et crée la base si elle n'existe pas.	1828
<u>8.124.39 sqlite_prev()</u> : SQLiteResult->prev().	1829
<u>8.124.40 sqlite_query()</u> : SQLiteDatabase->query().	1829
<u>8.124.41 sqlite_rewind()</u> : SQLiteResult->rewind().	1829
<u>8.124.42 sqlite_seek()</u> : SQLiteResult->seek().	1829
<u>8.124.43 sqlite_single_query()</u> : SQLiteDatabase->singleQuery().	1830
<u>8.124.44 sqlite_udf_decode_binary()</u> : Décode des données binaires, passées à une UDF SQLite.	1830
<u>8.124.45 sqlite_udf_encode_binary()</u> : Encode les données binaires d'une UDF SQLite avant de les retourner.	1830
<u>8.124.46 sqlite_unbuffered_query()</u> : SQLiteDatabase->unbufferedQuery().	1830
<u>8.124.47 sqlite_valid()</u> : SQLiteResult->valid().	1831
<u>8.125 Shell2 sécurisé.</u>	1832
<u>8.125.1 Introduction.</u>	1832
<u>8.125.2 Installation.</u>	1832
<u>8.125.3 Constantes pré-définies.</u>	1833
<u>8.125.5 ssh2_auth_none()</u> : Identification en tant que "none".	1834
<u>8.125.6 ssh2_auth_password()</u> : Identification via SSH en utilisant un mot de passe en clair.	1834
<u>8.125.7 ssh2_auth_pubkey_file()</u> : Identification en utilisant une clé publique.	1835
<u>8.125.8 ssh2_connect()</u> : Connexion à un serveur SSH.	1835
<u>8.125.9 ssh2_exec()</u> : Exécute une commande sur un serveur distant.	1837
<u>8.125.10 ssh2_fetch_stream()</u> : Parcours un flux étendu de données.	1837
<u>8.125.11 ssh2_fingerprint()</u> : Récupère l'empreinte d'un serveur distant.	1837
<u>8.125.12 ssh2_methods_negotiated()</u> : Retourne une liste de méthodes négociées.	1838
<u>8.125.13 ssh2_publickey_add()</u> : Ajoute une clé publique autorisée.	1838
<u>8.125.14 ssh2_publickey_init()</u> : Initialise un Publickey Subsystem (sous-système de clé publique).	1839
<u>8.125.15 ssh2_publickey_list()</u> : Liste les clés publiques actuellement autorisées.	1839

Sommaire

8 Référence des fonctions

<u>8.125.16 ssh2_publickey_remove()</u> : Supprime un clé publique autorisée.....	1839
<u>8.125.17 ssh2_scp_recv()</u> : Demande un fichier via SCP.....	1839
<u>8.125.18 ssh2_scp_send()</u> : Envoie un fichier via SCP.....	1840
<u>8.125.19 ssh2_sftp_lstat()</u> : Statue un lien symbolique.....	1840
<u>8.125.20 ssh2_sftp_mkdir()</u> : Crée un dossier.....	1840
<u>8.125.21 ssh2_sftp_readlink()</u> : Retourne la cible d'un lien symbolique.....	1841
<u>8.125.22 ssh2_sftp_realpath()</u> : Résoud le chemin réel d'un chemin fourni.....	1841
<u>8.125.23 ssh2_sftp_rename()</u> : Renomme un fichier distant.....	1842
<u>8.125.24 ssh2_sftp_rmdir()</u> : Efface un dossier.....	1842
<u>8.125.25 ssh2_sftp_stat()</u> : Statue un fichier sur un système de fichiers distant.....	1842
<u>8.125.26 ssh2_sftp_symlink()</u> : Crée un lien symbolique.....	1843
<u>8.125.27 ssh2_sftp_unlink()</u> : Efface un fichier.....	1843
<u>8.125.28 ssh2_sftp()</u> : Initialise un sous-système SFTP.....	1843
<u>8.125.29 ssh2_shell()</u> : Demande un shell interactif.....	1844
<u>8.125.30 ssh2_tunnel()</u> : Ouvre un tunnel à travers un serveur distant.....	1844
<u>8.126 Flux</u>	1846
<u>8.126.1 Introduction</u>	1846
<u>8.126.2 Filtres de flux</u>	1846
<u>8.126.3 Contextes de flux</u>	1846
<u>8.126.4 Installation</u>	1847
<u>8.126.5 Classes Stream</u>	1847
<u>8.126.6 Constantes pré-définies</u>	1847
<u>8.126.7 Erreurs de flux</u>	1849
<u>8.126.8 Exemples</u>	1849
<u>8.126.10 stream_bucket_make_writeable()</u> : Retourne un objet de compartiment depuis le corps pour des opérations sur celui-ci.....	1851
<u>8.126.11 stream_bucket_new()</u> : Crée un nouveau compartiment pour l'utiliser sur le flux courant.....	1851
<u>8.126.12 stream_bucket_prepend()</u> : Ajout initial d'un compartiment au corps.....	1852
<u>8.126.13 stream_context_create()</u> : Crée un contexte de flux.....	1852
<u>8.126.14 stream_context_get_default()</u> : Lit le contexte par défaut des flux.....	1852
<u>8.126.15 stream_context_get_options()</u> : Lit la valeur des options pour un flux/gestionnaire/contexte.....	1853
<u>8.126.16 stream_context_set_option()</u> : Configure une option pour un flux/gestionnaire/contexte.....	1853
<u>8.126.17 stream_context_set_params()</u> : Configure les paramètres pour un flux/gestionnaire/contexte.....	1854
<u>8.126.18 stream_copy_to_stream()</u> : Copie des données depuis un flux vers un autre.....	1854
<u>8.126.19 stream_filter_append()</u> : Attache un filtre à un flux en fin de liste.....	1854
<u>8.126.20 stream_filter_prepend()</u> : Attache un filtre à un flux en début de liste.....	1855
<u>8.126.21 stream_filter_register()</u> : Enregistre un filtre de flux.....	1856
<u>8.126.22 stream_filter_remove()</u> : Supprime un filtre d'un flux.....	1859
<u>8.126.23 stream_get_contents()</u> : Lit le reste d'un flux dans une chaîne.....	1859
<u>8.126.24 stream_get_filters()</u> : Liste les filtres enregistrés.....	1860
<u>8.126.25 stream_get_line()</u> : Lit une ligne dans un flux.....	1860
<u>8.126.26 stream_get_meta_data()</u> : Lit les en-têtes et données méta des flux.....	1860
<u>8.126.27 stream_get_transports()</u> : Liste les gestionnaires de transports de sockets disponibles.....	1861
<u>8.126.28 stream_get_wrappers()</u> : Liste les gestionnaires de flux.....	1862
<u>8.126.29 stream_register_wrapper()</u> : Alias de stream_wrapper_register.....	1863

Sommaire

8 Référence des fonctions

<u>8.126.30 stream_select()</u> : Retourne l'équivalent de l'appel système select() sur un tableau de flux avec un délai d'expiration spécifié par tv_sec et tv_usec.....	1863
<u>8.126.31 stream_set_blocking()</u> : Configure le mode bloquant d'un flux.....	1865
<u>8.126.32 stream_set_timeout()</u> : Configure la durée d'expiration d'un flux.....	1865
<u>8.126.33 stream_set_write_buffer()</u> : Configure la bufferisation de fichier pour un flux.....	1866
<u>8.126.34 stream_socket_accept()</u> : Accepte une connexion sur une socket créée par stream_socket_server.....	1866
<u>8.126.35 stream_socket_client()</u> : Ouvre une connexion socket Internet ou Unix....	1867
<u>8.126.36 stream_socket_enable_crypto()</u> : Active ou non le cryptage sur une socket déjà connectée.....	1868
<u>8.126.37 stream_socket_get_name()</u> : Lit le nom des sockets locale ou distante.....	1869
<u>8.126.38 stream_socket_pair()</u> : Crée une paire de sockets connectées et indissociables.....	1869
<u>8.126.39 stream_socket_recvfrom()</u> : Lit des données depuis une socket connectée ou pas.....	1869
<u>8.126.40 stream_socket_sendto()</u> : Envoie un message à la socket, connectée ou pas.....	1870
<u>8.126.41 stream_socket_server()</u> : Crée une socket serveur Unix ou Internet.....	1871
<u>8.126.42 stream_wrapper_register()</u> : Enregistre une enveloppe URL implémentée comme une classe PHP.....	1872
<u>8.126.43 stream_wrapper_restore()</u> : Restaure un gestionnaire d'URL supprimé.....	1877
<u>8.126.44 stream_wrapper_unregister()</u> : Supprime un gestionnaire d'URL.....	1877
<u>8.127 Chaînes de caractères (Strings)</u>	1878
<u>8.127.1 Introduction</u>	1878
<u>8.127.2 Pré-requis</u>	1878
<u>8.127.3 Installation</u>	1878
<u>8.127.4 Constantes pré-définies</u>	1878
<u>8.127.5 Voir aussi</u>	1879
<u>8.127.7 addslashes()</u> : Ajoute des anti-slash dans une chaîne.....	1881
<u>8.127.8 bin2hex()</u> : Convertit des données binaires en représentation hexadécimale.....	1881
<u>8.127.9 chop()</u> : Alias de rtrim.....	1881
<u>8.127.10 chr()</u> : Retourne un caractère spécifique.....	1881
<u>8.127.11 chunk_split()</u> : Scinde une chaîne.....	1882
<u>8.127.12 convert_cyr_string()</u> : Convertit une chaîne d'un jeu de caractères cyrillique à l'autre.....	1882
<u>8.127.13 convert_uudecode()</u> : Décode une chaîne au format uuencode.....	1883
<u>8.127.14 convert_uuencode()</u> : Encode une chaîne de caractères en utilisant l'algorithme uuencode.....	1883
<u>8.127.15 count_chars()</u> : Retourne des statistiques sur les caractères utilisés dans une chaîne.....	1883
<u>8.127.16 crc32()</u> : Calcule la somme de contrôle CRC32.....	1884
<u>8.127.17 crypt()</u> : Chiffre indéchiffrable (hashing).....	1884
<u>8.127.18 echo()</u> : Affiche une chaîne de caractères.....	1886
<u>8.127.19 explode()</u> : Coupe une chaîne en segments.....	1887
<u>8.127.20 fprintf()</u> : Écrit une chaîne formatée dans un flux.....	1889
<u>8.127.21 get_html_translation_table()</u> : Retourne la table de traduction des entités utilisée par htmlspecialchars et htmlentities.....	1889
<u>8.127.22 hebrew()</u> : Convertit un texte logique hébreu en texte visuel.....	1889
<u>8.127.23 hebrewc()</u> : Convertit un texte logique hébreu en texte visuel, avec	

Sommaire

8 Référence des fonctions

<u>retours à la ligne</u>	1890
<u>8.127.24 html_entity_decode() : Convertit toutes les entités HTML en caractères normaux</u>	1890
<u>8.127.25 htmlentities() : Convertit tous les caractères éligibles en entités HTML</u>	1891
<u>8.127.26 htmlspecialchars_decode() : Convertit les entités HTML spéciales en caractères</u>	1893
<u>8.127.27 htmlspecialchars() : Convertit les caractères spéciaux en entités HTML</u>	1893
<u>8.127.28 implode() : Rassemble les éléments d'un tableau en une chaîne</u>	1894
<u>8.127.29 join() : Alias de implode</u>	1895
<u>8.127.30 levenshtein() : Calcule la distance Levenshtein entre deux chaînes</u>	1895
<u>8.127.31 localeconv() : Lit la configuration locale</u>	1896
<u>8.127.32 ltrim() : Supprime les espaces (ou d'autres caractères) de début de chaîne</u>	1898
<u>8.127.33 md5_file() : Calcule le md5 d'un fichier</u>	1898
<u>8.127.34 md5() : Calcule le md5 d'une chaîne</u>	1899
<u>8.127.35 metaphone() : Calcule la clé metaphone</u>	1899
<u>8.127.36 money_format() : Met un nombre au format monétaire</u>	1899
<u>8.127.37 nl_langinfo() : Rassemble des informations sur la langue et la configuration locale</u>	1902
<u>8.127.38 nl2br() : Insère un retour à la ligne HTML à chaque nouvelle ligne</u>	1904
<u>8.127.39 number_format() : Formate un nombre pour l'affichage</u>	1904
<u>8.127.40 ord() : Retourne le code ASCII d'un caractère</u>	1905
<u>8.127.41 parse_str() : Analyse une requête HTTP</u>	1905
<u>8.127.42 print() : Affiche une chaîne de caractères</u>	1906
<u>8.127.43 printf() : Affiche une chaîne de caractères formatée</u>	1907
<u>8.127.44 quoted_printable_decode() : Convertit une chaîne quoted-printable en chaîne 8 bits</u>	1907
<u>8.127.45 quotemeta() : Echappe les méta-caractères</u>	1907
<u>8.127.46 rtrim() : Supprime les espaces (ou d'autres caractères) de fin de chaîne</u>	1908
<u>8.127.47 setlocale() : Modifie les informations de localisation</u>	1908
<u>8.127.48 sha1_file() : Calcule le sha1 d'un fichier</u>	1910
<u>8.127.49 sha1() : Calcule le sha1 d'une chaîne de caractères</u>	1910
<u>8.127.50 similar_text() : Calcule la similarité de deux chaînes</u>	1910
<u>8.127.51 soundex() : Calcule la clé soundex</u>	1911
<u>8.127.52 sprintf() : Retourne une chaîne formatée</u>	1911
<u>8.127.53 sscanf() : Analyse une chaîne à l'aide d'un format</u>	1913
<u>8.127.54 str_ireplace() : Version insensible à la casse de str_replace</u>	1914
<u>8.127.55 str_pad() : Complète une chaîne jusqu'à une taille donnée</u>	1914
<u>8.127.56 str_repeat() : Répète une chaîne</u>	1915
<u>8.127.57 str_replace() : Remplace toutes les occurrences dans une chaîne</u>	1915
<u>8.127.58 str_rot13() : Effectue une transformation ROT13</u>	1916
<u>8.127.59 str_shuffle() : Mélange les caractères d'une chaîne de caractères</u>	1916
<u>8.127.60 str_split() : Convertit une chaîne de caractères en tableau</u>	1917
<u>8.127.61 str_word_count() : Compte le nombre de mots utilisés dans une chaîne</u>	1918
<u>8.127.62 strcasecmp() : Comparaison insensible à la casse de chaînes binaires</u>	1918
<u>8.127.63 strchr() : Alias de strpos</u>	1919
<u>8.127.64 strcmp() : Comparaison binaire de chaînes</u>	1919
<u>8.127.65 strcoll() : Comparaison de chaînes localisées</u>	1919
<u>8.127.66 strpos() : Trouve un segment de chaîne ne contenant pas certains caractères</u>	1919
<u>8.127.67 strip_tags() : Supprime les balises HTML et PHP d'une chaîne</u>	1920

Sommaire

8 Référence des fonctions

<u>8.127.68 stripslashes()</u> : Décode une chaîne encodée avec addcslashes.....	1920
<u>8.127.69 stripos()</u> : Recherche la première occurrence dans une chaîne, sans tenir compte de la casse.....	1921
<u>8.127.70 stripslashes()</u> : Supprime les anti-slash d'une chaîne.....	1921
<u>8.127.71 strtolower()</u> : Version insensible à la casse de strtolower.....	1923
<u>8.127.72 strlen()</u> : Calcule la taille d'une chaîne.....	1923
<u>8.127.73 strnatcasecmp()</u> : Comparaison de chaînes avec l'algorithme d'ordre naturel" (insensible à la casse).....	1924
<u>8.127.74 strnatcmp()</u> : Comparaison de chaînes avec l'algorithme d'ordre naturel".....	1924
<u>8.127.75 strncasecmp()</u> : Compare en binaire des chaînes de caractères.....	1925
<u>8.127.76 strncmp()</u> : Comparaison binaire des n premiers caractères.....	1925
<u>8.127.77 strpos()</u> : Recherche une chaîne de caractères dans un ensemble de caractères.....	1926
<u>8.127.78 strrpos()</u> : Trouve la position d'un caractère dans une chaîne.....	1926
<u>8.127.79 strrchr()</u> : Trouve la dernière occurrence d'un caractère dans une chaîne.....	1927
<u>8.127.80 strrev()</u> : Inverse une chaîne.....	1927
<u>8.127.81 stripos()</u> : Trouve la position de la dernière occurrence d'une chaîne dans une autre de façon insensible à la casse.....	1928
<u>8.127.82 strrpos()</u> : Trouve la position de la dernière occurrence d'un caractère dans une chaîne.....	1928
<u>8.127.83 strstr()</u> : Trouve le premier segment de chaîne.....	1929
<u>8.127.84 strpos()</u> : Trouve la première occurrence dans une chaîne.....	1930
<u>8.127.85 strtok()</u> : Coupe une chaîne en segments.....	1930
<u>8.127.86 strtolower()</u> : Renvoie une chaîne en minuscules.....	1932
<u>8.127.87 strtoupper()</u> : Renvoie une chaîne en majuscules.....	1932
<u>8.127.88 strstr()</u> : Remplace des caractères dans une chaîne.....	1933
<u>8.127.89 substr_compare()</u> : Compare deux chaînes depuis un offset jusqu'à une longueur en caractères.....	1933
<u>8.127.90 substr_count()</u> : Compte le nombre d'occurrences de segments dans une chaîne.....	1934
<u>8.127.91 substr_replace()</u> : Remplace un segment dans une chaîne.....	1934
<u>8.127.92 substr()</u> : Retourne un segment de chaîne.....	1935
<u>8.127.93 trim()</u> : Supprime les espaces (ou d'autres caractères) en début et fin de chaîne.....	1936
<u>8.127.94 ucfirst()</u> : Met le premier caractère en majuscule.....	1936
<u>8.127.95 ucwords()</u> : Met en majuscule la première lettre de tous les mots.....	1936
<u>8.127.96 vfprintf()</u> : Écrit une chaîne formatée dans un flux.....	1937
<u>8.127.97 fprintf()</u> : Affiche une chaîne formatée.....	1937
<u>8.127.98 sprintf()</u> : Retourne une chaîne formatée.....	1937
<u>8.127.99 wordwrap()</u> : Effectue la césure d'une chaîne.....	1938
<u>8.128 Shockwave Flash</u>	1939
<u>8.128.1 Introduction</u>	1939
<u>8.128.2 Pré-requis</u>	1939
<u>8.128.3 Installation</u>	1939
<u>8.128.4 Configuration à l'exécution</u>	1939
<u>8.128.5 Types de ressources</u>	1939
<u>8.128.6 Constantes pré-définies</u>	1939
<u>8.128.7 Exemples</u>	1940
<u>8.128.9 swf_actiongotoframe()</u> : Joue un cadre puis stoppe.....	1942

Sommaire

8 Référence des fonctions

<u>8.128.10 swf_actiongotolabel()</u> : Joue le cadre Flash spécifié.....	1942
<u>8.128.11 swf_actionnextframe()</u> : Avance d'un cadre.....	1942
<u>8.128.12 swf_actionplay()</u> : Joue l'animation Flash à partir du cadre courant.....	1943
<u>8.128.13 swf_actionprevframe()</u> : Recule d'un cadre.....	1943
<u>8.128.14 swf_actionsettarget()</u> : Fixe le contexte des actions.....	1943
<u>8.128.15 swf_actionstop()</u> : Arrête l'animation Flash.....	1943
<u>8.128.16 swf_actiontogglequality()</u> : Choisit le niveau de qualité de l'animation Flash.....	1943
<u>8.128.17 swf_actionwaitforframe()</u> : Ignore les actions si le cadre Flash n'est pas chargé.....	1943
<u>8.128.18 swf_addbuttonrecord()</u> : Contrôle la situation, l'apparence et la zone active du bouton Flash courant.....	1943
<u>8.128.19 swf_addcolor()</u> : Fixe la couleur globale d'addition Flash (? : the global add color).....	1944
<u>8.128.20 swf_closefile()</u> : Ferme le fichier courant Shockwave Flash.....	1944
<u>8.128.21 swf_definebitmap()</u> : Définit une image bitmap.....	1945
<u>8.128.22 swf_definefont()</u> : Définit une police Flash.....	1945
<u>8.128.23 swf_defineline()</u> : Définit une ligne.....	1945
<u>8.128.24 swf_definepoly()</u> : Définit un polygone Flash.....	1945
<u>8.128.25 swf_definerect()</u> : Définit un rectangle.....	1946
<u>8.128.26 swf_definetext()</u> : Définit une chaîne de texte.....	1946
<u>8.128.27 swf_endbutton()</u> : Termine la définition du bouton Flash courant.....	1946
<u>8.128.28 swf_enddoaction()</u> : Termine l'action courante.....	1946
<u>8.128.29 swf_endshape()</u> : Complète la définition de la forme Flash courante.....	1946
<u>8.128.30 swf_endsymbol()</u> : Termine la définition de symbole.....	1946
<u>8.128.31 swf_fontsize()</u> : Change la taille de la police.....	1947
<u>8.128.32 swf_fontslant()</u> : Change l'inclinaison de la police courante.....	1947
<u>8.128.33 swf_fontracking()</u> : Change l'espacement des caractères.....	1947
<u>8.128.34 swf_getbitmapinfo()</u> : Lit les informations sur une image.....	1947
<u>8.128.35 swf_getfontinfo()</u> : Retourne la hauteur du A majuscule, et du x minuscule.....	1947
<u>8.128.36 swf_getframe()</u> : Retourne le numéro de cadre courant.....	1947
<u>8.128.37 swf_labelframe()</u> : Nomme le cadre courant.....	1948
<u>8.128.38 swf_lookat()</u> : Définit une transformation de vue.....	1948
<u>8.128.39 swf_modifyobject()</u> : Modifie un objet Flash.....	1948
<u>8.128.40 swf_mulcolor()</u> : Fixe la couleur globale de multiplication (? : the global multiply color).....	1948
<u>8.128.41 swf_nextid()</u> : Retourne le prochain identifiant d'objet libre.....	1948
<u>8.128.42 swf_oncondition()</u> : Décrit la transition utilisée pour déclencher une liste d'actions.....	1949
<u>8.128.43 swf_openfile()</u> : Ouvre un nouveau fichier Shockwave Flash.....	1949
<u>8.128.44 swf_ortho()</u> : Définit une projection orthogonale entre les coordonnées utilisateur et le port courant.....	1949
<u>8.128.45 swf_ortho2()</u> : Définit une projection orthogonale à 2 dimensions entre les coordonnées utilisateur et le port courant.....	1950
<u>8.128.46 swf_perspective()</u> : Définit une projection orthogonale à 3 dimensions entre les coordonnées utilisateur et le port courant.....	1950
<u>8.128.47 swf_placeobject()</u> : Place un objet sur la scène.....	1950
<u>8.128.48 swf_polarview()</u> : Définit le point de vue de l'utilisateur en coordonnées polaires.....	1950
<u>8.128.49 swf_popmatrix()</u> : Dépèle la matrice de transformation.....	1951

Sommaire

8 Référence des fonctions

<u>8.128.50 swf_posround()</u> : Active l'approximation des translations d'objets Flash....	1951
<u>8.128.51 swf_pushmatrix()</u> : Empile la matrice de transformations courante dans la pile.....	1951
<u>8.128.52 swf_removeobject()</u> : Enlève un objet.....	1951
<u>8.128.53 swf_rotate()</u> : Rotation de la transformation courante.....	1951
<u>8.128.54 swf_scale()</u> : Homothétie.....	1951
<u>8.128.55 swf_setfont()</u> : Change la police courante.....	1951
<u>8.128.56 swf_setframe()</u> : Fixe le cadre courant.....	1952
<u>8.128.57 swf_shapearc()</u> : Dessine un arc de cercle.....	1952
<u>8.128.58 swf_shapecurveto()</u> : Dessine une courbe de Bézier quadratique entre deux points.....	1952
<u>8.128.59 swf_shapecurveto3()</u> : Dessine une courbe Bézier cubique.....	1952
<u>8.128.60 swf_shapefillbitmapclip()</u> : Choisit le mode de remplissage Flash par texture.....	1952
<u>8.128.61 swf_shapefillbitmaptile()</u> : Choisit le mode de remplissage par texture Flash répétée.....	1952
<u>8.128.62 swf_shapefilloff()</u> : Désactive le remplissage de formes Flash.....	1953
<u>8.128.63 swf_shapefillsolid()</u> : Fixe la couleur pour le style courant de remplissage Flash.....	1953
<u>8.128.64 swf_shapelinesolid()</u> : Fixe le style courant de ligne.....	1953
<u>8.128.65 swf_shapelineto()</u> : Dessine une ligne.....	1953
<u>8.128.66 swf_shapemoveto()</u> : Change la position courante.....	1953
<u>8.128.67 swf_showframe()</u> : Affiche le cadre courant.....	1953
<u>8.128.68 swf_startbutton()</u> : Commence la définition d'un bouton.....	1953
<u>8.128.69 swf_startdoaction()</u> : Commence la description d'une liste d'actions pour le cadre courant.....	1954
<u>8.128.70 swf_startshape()</u> : Commence une forme complexe.....	1954
<u>8.128.71 swf_startsymbol()</u> : Définit un symbole.....	1954
<u>8.128.72 swf_textwidth()</u> : Retourne la longueur d'une chaîne.....	1954
<u>8.128.73 swf_translate()</u> : Translate la transformation courante.....	1954
<u>8.128.74 swf_viewport()</u> : Sélectionne une nouvelle zone pour un dessin ultérieur.....	1954
<u>8.129 Sybase</u>	1955
<u>8.129.1 Introduction</u>	1955
<u>8.129.2 Pré-requis</u>	1955
<u>8.129.3 Installation</u>	1955
<u>8.129.4 Configuration à l'exécution</u>	1955
<u>8.129.5 Types de ressources</u>	1957
<u>8.129.6 Constantes pré-définies</u>	1957
<u>8.129.8 sybase_close()</u> : Ferme une connexion Sybase.....	1958
<u>8.129.9 sybase_connect()</u> : Ouvre une connexion à un serveur Sybase.....	1958
<u>8.129.10 sybase_data_seek()</u> : Déplace le pointeur interne de lignes Sybase.....	1958
<u>8.129.11 sybase_deadlock_retry_count()</u> : Configure le nombre de tentatives lors de blocages (deadlock).....	1959
<u>8.129.12 sybase_fetch_array()</u> : Retourne une ligne Sybase sous la forme d'un tableau.....	1959
<u>8.129.13 sybase_fetch_assoc()</u> : Lit une ligne de résultat Sybase sous forme de tableau associatif.....	1960
<u>8.129.14 sybase_fetch_field()</u> : Lit les informations d'un champ Sybase.....	1960
<u>8.129.15 sybase_fetch_object()</u> : Retourne une ligne Sybase sous la forme d'un objet.....	1961

Sommaire

8 Référence des fonctions

8.129.16 <u>sybase_fetch_row()</u> : Retourne une ligne Sybase sous la forme d'un tableau numérique.....	1962
8.129.17 <u>sybase_field_seek()</u> : Modifie l'index d'un champ Sybase.....	1962
8.129.18 <u>sybase_free_result()</u> : Libère un résultat Sybase de la mémoire.....	1962
8.129.19 <u>sybase_get_last_message()</u> : Retourne le dernier message du serveur.....	1963
8.129.20 <u>sybase_min_client_severity()</u> : Fixe la sévérité minimale du client Sybase.....	1963
8.129.21 <u>sybase_min_error_severity()</u> : Fixe la sévérité minimale du client pour les erreurs.....	1963
8.129.22 <u>sybase_min_message_severity()</u> : Fixe la sévérité minimale du client pour les messages.....	1963
8.129.23 <u>sybase_min_server_severity()</u> : Fixe la sévérité minimale du client pour le serveur Sybase.....	1964
8.129.24 <u>sybase_num_fields()</u> : Retourne le nombre de champs dans un résultat Sybase.....	1964
8.129.25 <u>sybase_num_rows()</u> : Retourne le nombre de lignes dans un résultat Sybase.....	1964
8.129.26 <u>sybase_pconnect()</u> : Ouvre une connexion persistante à un serveur Sybase.....	1964
8.129.27 <u>sybase_query()</u> : Envoie une requête à une base Sybase.....	1965
8.129.28 <u>sybase_result()</u> : Lit une valeur dans un résultat.....	1965
8.129.29 <u>sybase_select_db()</u> : Sélectionne une base de données Sybase.....	1965
8.129.30 <u>sybase_set_message_handler()</u> : Configure le gestionnaire de messages Sybase.....	1966
8.129.31 <u>sybase_unbuffered_query()</u> : Envoie une requête à Sybase et ne bloque pas.....	1967
8.130 <u>Fonctions TCP Wrappers</u>	1968
8.130.1 <u>Introduction</u>	1968
8.130.2 <u>Installation</u>	1968
8.131 <u>Tidy</u>	1969
8.131.1 <u>Introduction</u>	1969
8.131.2 <u>Pré-requis</u>	1969
8.131.3 <u>Installation</u>	1969
8.131.4 <u>Configuration à l'exécution</u>	1969
8.131.5 <u>Types de ressources</u>	1970
8.131.6 <u>Classes pré-définies</u>	1970
8.131.7 <u>Constantes pré-définies</u>	1970
8.131.8 <u>Exemples</u>	1977
8.131.10 <u>tidy_access_count()</u> : Retourne le nombre d'alertes d'accessibilité Tidy rencontrées dans le document.....	1979
8.131.11 <u>tidy_clean_repair()</u> : Effectue les opérations de nettoyage et de réparation préparées pour un fichier HTML.....	1979
8.131.12 <u>tidy_config_count()</u> : Retourne le nombre d'erreurs de configuration Tidy rencontrées dans le document.....	1980
8.131.13 <u>tidy::construct()</u> : Construit un nouvel objet Tidy.....	1980
8.131.14 <u>tidy_diagnose()</u> : Etablit le diagnostic pour le document analysé et réparé.....	1981
8.131.15 <u>tidy_error_count()</u> : Retourne le nombre d'erreurs Tidy rencontrées dans le document.....	1982
8.131.16 <u>tidy_get_body()</u> : Retourne un objet TidyNode, commencé à partir de la balise <body>.....	1983

Sommaire

8 Référence des fonctions

<u>8.131.17 tidy_get_config() : Lit la configuration Tidy courante</u>	1984
<u>8.131.18 tidy_get_error_buffer() : Retourne les alertes et erreurs qui sont survenues lors de l'analyse du document</u>	1986
<u>8.131.19 tidy_get_head() : Retourne un objet TidyNode à partir de la balise <head></u>	1986
<u>8.131.20 tidy_get_html_ver() : Détecte le version du code HTML utilisée dans un document</u>	1987
<u>8.131.21 tidy_get_html() : Retourne un objet TidyNode commençant à la balise <html></u>	1987
<u>8.131.22 tidy_get_opt_doc() : Retourne la documentation pour le nom de l'option donnée</u>	1988
<u>8.131.23 tidy_get_output() : Retourne une chaîne représentant les balises telles qu'analysées par Tidy</u>	1989
<u>8.131.24 tidy_get_release() : Retourne la date de publication (version) de la bibliothèque Tidy</u>	1989
<u>8.131.25 tidy_get_root() : Retourne un objet tidyNode représentant la racine du document HTML</u>	1989
<u>8.131.26 tidy_get_status() : Retourne le statut du document spécifié</u>	1990
<u>8.131.27 tidy_getopt() : Retourne la valeur de l'option de configuration Tidy</u>	1991
<u>8.131.28 tidy_is_xhtml() : Indique si le document est un document XHTML</u>	1992
<u>8.131.29 tidy_is_xml() : Indique si le document est un document XML générique (non HTML/XHTML)</u>	1992
<u>8.131.30 tidy_load_config() : Charge un fichier de configuration ASCII Tidy avec l'encodage spécifié</u>	1992
<u>8.131.31 tidy_node->get_attr() : Retourne la valeur de l'attribut spécifié</u>	1993
<u>8.131.32 tidy_node->get_nodes() : Retourne un tableau avec les noeuds placés sous le noeud courant, avec l'identifiant spécifié</u>	1993
<u>8.131.33 tidy_node->next() : Retourne le prochain frère du noeud courant</u>	1993
<u>8.131.34 tidy_node->prev() : Retourne le frère précédent de ce noeud</u>	1993
<u>8.131.35 tidy_parse_file() : Analyse les balises d'un fichier ou d'une URI</u>	1993
<u>8.131.36 tidy_parse_string() : Analyse un document HTML contenu dans une chaîne</u>	1994
<u>8.131.37 tidy_repair_file() : Répare un fichier et le renvoie en tant que chaîne</u>	1995
<u>8.131.38 tidy_repair_string() : Répare une chaîne HTML en utilisant un fichier de configuration optionnel</u>	1996
<u>8.131.39 tidy_reset_config() : Redonne les valeurs de configuration par défaut de Tidy</u>	1996
<u>8.131.40 tidy_save_config() : Sauve la configuration courante dans un fichier</u>	1997
<u>8.131.41 tidy_set_encoding() : Modifie le jeu de caractères pour les entrées/sorties de l'analyseur Tidy</u>	1997
<u>8.131.42 tidy_setopt() : Modifie la valeur de l'option de configuration Tidy</u>	1997
<u>8.131.43 tidy_warning_count() : Retourne le nombre d'alertes Tidy rencontrées dans le document spécifié</u>	1998
<u>8.131.44 tidyNode->hasChildren() : Retourne true si le noeud a des enfants</u>	1998
<u>8.131.45 tidyNode->hasSiblings() : Retourne true si le noeud a des frères</u>	1998
<u>8.131.46 tidyNode->isAsp() : Retourne TRUE si ce noeud Tidy est du code ASP</u>	1998
<u>8.131.47 tidyNode->isComment() : Retourne true si le noeud représente un commentaire</u>	1999
<u>8.131.48 tidyNode->isHtml() : Retourne true si le noeud est une partie d'un document HTML</u>	1999
<u>8.131.49 tidyNode->isJste() : Renvoie true: si ce node est JSTE</u>	1999

Sommaire

8 Référence des fonctions

<u>8.131.50 tidyNode->isPhp()</u> : Retourne TRUE si ce noeud est en PHP.....	1999
<u>8.131.51 tidyNode->isText()</u> : Retourne true si le noeud représente du texte (aucun marquage).....	2000
8.132 Analyseur de code PHP	2001
<u>8.132.1 Introduction</u>	2001
<u>8.132.2 Pré-requis</u>	2001
<u>8.132.3 Installation</u>	2001
<u>8.132.4 Constantes pré-définies</u>	2001
<u>8.132.5 Exemples</u>	2001
<u>8.132.7 token_name()</u> : Lit le nom d'un élément de code source.....	2002
8.133 Fonctions ODBC (unifiés)	2003
<u>8.133.1 Introduction</u>	2003
<u>8.133.2 Pré-requis</u>	2003
<u>8.133.3 Installation</u>	2003
<u>8.133.4 Configuration à l'exécution</u>	2004
<u>8.133.5 Types de ressources</u>	2005
<u>8.133.6 Constantes pré-définies</u>	2005
<u>8.133.8 odbc_binmode()</u> : Modifie la gestion des colonnes de données binaires.....	2007
<u>8.133.9 odbc_close_all()</u> : Ferme toutes les connexions ODBC.....	2008
<u>8.133.10 odbc_close()</u> : Ferme une connexion ODBC.....	2008
<u>8.133.11 odbc_columnprivileges()</u> : Liste les colonnes et leurs droits associés.....	2008
<u>8.133.12 odbc_columns()</u> : Liste les colonnes d'une table.....	2009
<u>8.133.13 odbc_commit()</u> : Valide une transaction ODBC.....	2009
<u>8.133.14 odbc_connect()</u> : Connexion à une source.....	2009
<u>8.133.15 odbc_cursor()</u> : Lecture du pointeur de fiche courante (cursorname).....	2010
<u>8.133.16 odbc_data_source()</u> : Retourne des informations sur la connexion courante.....	2010
<u>8.133.17 odbc_do()</u> : Synonyme de odbc_exec.....	2010
<u>8.133.18 odbc_error()</u> : Lit le dernier code d'erreur.....	2011
<u>8.133.19 odbc_errormsg()</u> : Lit le dernier message d'erreur.....	2011
<u>8.133.20 odbc_exec()</u> : Prépare et exécute une requête SQL.....	2011
<u>8.133.21 odbc_execute()</u> : Exécute une requête SQL préparée.....	2011
<u>8.133.22 odbc_fetch_array()</u> : Lit une ligne de résultat dans un tableau associatif.....	2012
<u>8.133.23 odbc_fetch_into()</u> : Lit une ligne de résultat, et la place dans un tableau.....	2012
<u>8.133.24 odbc_fetch_object()</u> : Lit une ligne de résultat dans un objet.....	2013
<u>8.133.25 odbc_fetch_row()</u> : Lit une ligne de résultat.....	2013
<u>8.133.26 odbc_field_len()</u> : Lit la longueur d'un champ.....	2014
<u>8.133.27 odbc_field_name()</u> : Lit le nom de la colonne.....	2014
<u>8.133.28 odbc_field_num()</u> : Numéro de colonne.....	2014
<u>8.133.29 odbc_field_precision()</u> : Alias de odbc_field_len.....	2014
<u>8.133.30 odbc_field_scale()</u> : Lit l'échelle d'un champ.....	2014
<u>8.133.31 odbc_field_type()</u> : Type de données d'un champ.....	2014
<u>8.133.32 odbc_foreignkeys()</u> : Liste les clés étrangères.....	2015
<u>8.133.33 odbc_free_result()</u> : Libère les ressources associées à un résultat.....	2015
<u>8.133.34 odbc_gettypeinfo()</u> : Liste les types de données supportés par une source.....	2016
<u>8.133.35 odbc_longreadlen()</u> : Gestion des colonnes de type LONG.....	2016
<u>8.133.36 odbc_next_result()</u> : Vérifie si plusieurs résultats sont disponibles.....	2016
<u>8.133.37 odbc_num_fields()</u> : Nombre de colonnes dans un résultat.....	2017
<u>8.133.38 odbc_num_rows()</u> : Nombre de lignes dans un résultat.....	2017
<u>8.133.39 odbc_pconnect()</u> : Ouvre une connexion persistante à une source de	

Sommaire

8 Référence des fonctions

<u>données</u>	2017
<u>8.133.40 <code>odbc_prepare()</code> : Prépare une commande pour l'exécution</u>	2017
<u>8.133.41 <code>odbc_primarykeys()</code> : Liste les colonnes utilisées dans une clé primaire</u>	2018
<u>8.133.42 <code>odbc_procedurecolumns()</code> : Liste les paramètres des procédures</u>	2018
<u>8.133.43 <code>odbc_procedures()</code> : Liste les procédures stockées</u>	2019
<u>8.133.44 <code>odbc_result_all()</code> : Affiche le résultat sous la forme d'une table HTML</u>	2019
<u>8.133.45 <code>odbc_result()</code> : Lit un champ de résultat UODBC</u>	2019
<u>8.133.46 <code>odbc_rollback()</code> : Annule une transaction</u>	2020
<u>8.133.47 <code>odbc_setoption()</code> : Modifie les paramètres ODBC</u>	2020
<u>8.133.48 <code>odbc_specialcolumns()</code> : Retourne l'ensemble optimal de colonnes, qui permet de définir uniquement une ligne dans une table</u>	2021
<u>8.133.49 <code>odbc_statistics()</code> : Calcul des statistiques sur une table</u>	2021
<u>8.133.50 <code>odbc_tableprivileges()</code> : Liste les tables et leurs privilèges</u>	2022
<u>8.133.51 <code>odbc_tables()</code> : Liste les tables d'une source</u>	2022
<u>8.134 URL</u>	2024
<u>8.134.1 Introduction</u>	2024
<u>8.134.2 Pré-requis</u>	2024
<u>8.134.3 Installation</u>	2024
<u>8.134.4 Configuration à l'exécution</u>	2024
<u>8.134.5 Types de ressources</u>	2024
<u>8.134.6 Constantes pré-définies</u>	2024
<u>8.134.8 <code>base64_encode()</code> : Encode une chaîne en MIME base64</u>	2025
<u>8.134.9 <code>get_headers()</code> : Retourne tous les en-têtes envoyés par le serveur en réponse à une requête HTTP</u>	2025
<u>8.134.10 <code>get_meta_tags()</code> : Extrait toutes les balises méta d'un fichier HTML</u>	2026
<u>8.134.11 <code>http_build_query()</code> : Génère une chaîne de requête en encodage URL</u>	2027
<u>8.134.12 <code>parse_url()</code> : Analyse une URL et retourne ses composants</u>	2028
<u>8.134.13 <code>rawurldecode()</code> : Décode une chaîne URL</u>	2028
<u>8.134.14 <code>rawurlencode()</code> : Encode une chaîne en URL, selon la RFC 1738</u>	2029
<u>8.134.15 <code>urldecode()</code> : Décode une chaîne encodée URL</u>	2029
<u>8.134.16 <code>urlencode()</code> : Encode une chaîne en URL</u>	2030
<u>8.135 Fonctions de gestion des variables</u>	2031
<u>8.135.1 Introduction</u>	2031
<u>8.135.2 Pré-requis</u>	2031
<u>8.135.3 Installation</u>	2031
<u>8.135.4 Configuration à l'exécution</u>	2031
<u>8.135.5 Types de ressources</u>	2031
<u>8.135.6 Constantes pré-définies</u>	2031
<u>8.135.8 <code>doubleval()</code> : Alias de <code>floatval</code></u>	2032
<u>8.135.9 <code>empty()</code> : Détermine si une variable contient une valeur non nulle</u>	2032
<u>8.135.10 <code>floatval()</code> : Convertit une chaîne en nombre à virgule flottante</u>	2032
<u>8.135.11 <code>get_defined_vars()</code> : Liste toutes les variables définies</u>	2033
<u>8.135.12 <code>get_resource_type()</code> : Retourne le type de ressource</u>	2033
<u>8.135.13 <code>gettype()</code> : Retourne le type de la variable</u>	2033
<u>8.135.14 <code>import_request_variables()</code> : Importe les variables de GET/POST/Cookie dans l'environnement global</u>	2033
<u>8.135.15 <code>intval()</code> : Retourne la valeur numérique entière équivalent d'une variable</u>	2034
<u>8.135.16 <code>is_array()</code> : Détermine si une variable est un tableau</u>	2034
<u>8.135.17 <code>is_bool()</code> : Détermine si une variable est un booléen</u>	2035
<u>8.135.18 <code>is_callable()</code> : Détermine si l'argument peut être appelé comme fonction</u>	2035

Sommaire

8 Référence des fonctions

<u>8.135.19 is_double()</u> : Alias de <u>is_float</u>	2036
<u>8.135.20 is_float()</u> : Détermine si une variable est de type nombre décimal.....	2036
<u>8.135.21 is_int()</u> : Détermine si une variable est de type nombre entier.....	2036
<u>8.135.22 is_integer()</u> : Alias de <u>is_int</u>	2036
<u>8.135.23 is_long()</u> : Alias de <u>is_int</u>	2036
<u>8.135.24 is_null()</u> : Indique si une variable vaut NULL.....	2036
<u>8.135.25 is_numeric()</u> : Détermine si une variable est un type numérique.....	2037
<u>8.135.26 is_object()</u> : Détermine si une variable est de type objet.....	2037
<u>8.135.27 is_real()</u> : Alias de <u>is_float</u>	2037
<u>8.135.28 is_resource()</u> : Détermine si une variable est une ressource.....	2037
<u>8.135.29 is_scalar()</u> : Indique si une variable est un scalaire.....	2037
<u>8.135.30 is_string()</u> : Détermine si une variable est de type chaîne de caractères....	2037
<u>8.135.31 isset()</u> : Détermine si une variable est affectée.....	2038
<u>8.135.32 print_r()</u> : Affiche des informations lisibles pour une variable.....	2039
<u>8.135.33 serialize()</u> : Linéarise une variable.....	2040
<u>8.135.34 settype()</u> : Affecte un type à une variable.....	2041
<u>8.135.35 strval()</u> : Retourne la valeur de la variable, au format chaîne.....	2041
<u>8.135.36 unserialize()</u> : Crée une variable PHP à partir d'une valeur linéarisée.....	2042
<u>8.135.37 unset()</u> : Détruit une variable.....	2043
<u>8.135.38 var_dump()</u> : Affiche les informations d'une variable.....	2045
<u>8.135.39 var_export()</u> : Retourne le code PHP utilisé pour générer une variable.....	2045
<u>8.136 vpopmail</u>	2046
<u>8.136.1 Introduction</u>	2046
<u>8.136.2 Installation</u>	2046
<u>8.136.4 vpopmail_add_alias_domain()</u> : Ajout un alias pour un domaine virtuel.....	2046
<u>8.136.5 vpopmail_add_domain_ex()</u> : Ajoute un nouveau domaine virtuel.....	2047
<u>8.136.6 vpopmail_add_domain()</u> : Ajoute un nouveau domaine virtuel.....	2047
<u>8.136.7 vpopmail_add_user()</u> : Ajoute un nouvel utilisateur à un domaine virtuel....	2047
<u>8.136.8 vpopmail_alias_add()</u> : Ajoute un alias virtuel.....	2047
<u>8.136.9 vpopmail_alias_del_domain()</u> : Efface tous les alias virtuels d'un domaine.....	2048
<u>8.136.10 vpopmail_alias_del()</u> : Efface tous les alias d'un utilisateur.....	2048
<u>8.136.11 vpopmail_alias_get_all()</u> : Lit toutes les lignes d'alias d'un domaine.....	2048
<u>8.136.12 vpopmail_alias_get()</u> : Lit toutes les lignes d'un alias de domaine.....	2048
<u>8.136.13 vpopmail_auth_user()</u> : Valide le mot de passe d'un utilisateur pour un domaine.....	2049
<u>8.136.14 vpopmail_del_domain_ex()</u> : Efface un domaine virtuel.....	2049
<u>8.136.15 vpopmail_del_domain()</u> : Efface un domaine virtuel.....	2049
<u>8.136.16 vpopmail_del_user()</u> : Efface un utilisateur d'un domaine virtuel.....	2049
<u>8.136.17 vpopmail_error()</u> : Lit le dernier message d'erreur vpopmail.....	2050
<u>8.136.18 vpopmail_passwd()</u> : Remplace le mot de passe d'un utilisateur virtuel.....	2050
<u>8.136.19 vpopmail_set_user_quota()</u> : Modifie le quota d'un utilisateur virtuel.....	2050
<u>8.137 API windows</u>	2051
<u>8.137.1 Introduction</u>	2051
<u>8.137.2 Pré-requis</u>	2051
<u>8.137.3 Installation</u>	2051
<u>8.137.4 Configuration à l'exécution</u>	2051
<u>8.137.5 Types de ressources</u>	2051
<u>8.137.6 Constantes pré-définies</u>	2051
<u>8.137.7 Exemples</u>	2052
<u>8.137.9 w32api_init_dtype()</u> : Crée une instance de type de données Win32 et la	

Sommaire

8 Référence des fonctions

<u>remplit</u>	2053
<u>8.137.10 w32api_invoke_function()</u> : Appelle une fonction windows 32.....	2053
<u>8.137.11 w32api_register_function()</u> : Enregistre une fonction Win32 dans PHP.....	2053
<u>8.137.12 w32api_set_call_method()</u> : Modifie le type d'appel de la fonction Win32.....	2054
<u>8.138 WDDX</u>	2055
<u>8.138.1 Introduction</u>	2055
<u>8.138.2 Pré-requis</u>	2055
<u>8.138.3 Installation</u>	2055
<u>8.138.4 Configuration à l'exécution</u>	2055
<u>8.138.5 Types de ressources</u>	2055
<u>8.138.6 Constantes pré-définies</u>	2055
<u>8.138.7 Exemples</u>	2055
<u>8.138.9 wddx_deserialize()</u> : Alias de wddx_unserialize.....	2056
<u>8.138.10 wddx_packet_end()</u> : Clôt un paquet WDDX.....	2056
<u>8.138.11 wddx_packet_start()</u> : Commence un nouveau paquet WDDX avec une structure.....	2057
<u>8.138.12 wddx_serialize_value()</u> : Enregistre une valeur dans un paquet WDDX.....	2057
<u>8.138.13 wddx_serialize_vars()</u> : Enregistre plusieurs valeurs dans un paquet WDDX.....	2057
<u>8.138.14 wddx_unserialize()</u> : Délinéarise un paquet WDDX.....	2058
<u>8.139 Fonctions xattr</u>	2059
<u>8.139.1 Introduction</u>	2059
<u>8.139.2 Pré-requis</u>	2059
<u>8.139.3 Installation</u>	2059
<u>8.139.4 Constantes pré-définies</u>	2059
<u>8.139.6 xattr_list()</u> : Récupère une liste d'attributs étendus.....	2060
<u>8.139.7 xattr_remove()</u> : Efface un attribut étendu.....	2060
<u>8.139.8 xattr_set()</u> : Définit un attribut étendu.....	2060
<u>8.139.9 xattr_supported()</u> : Vérifie si un système de fichier supporte les attributs étendus.....	2060
<u>8.140 Fonctions xdiff</u>	2061
<u>8.140.1 Introduction</u>	2061
<u>8.140.2 Pré-requis</u>	2061
<u>8.140.3 Installation</u>	2061
<u>8.140.4 Constantes pré-définies</u>	2061
<u>8.140.6 xdiff_file_diff()</u> : Créé un diff unifié entre deux fichiers.....	2062
<u>8.140.7 xdiff_file_merge3()</u> : Fusionne trois fichiers en un seul.....	2062
<u>8.140.8 xdiff_file_patch_binary()</u> : Patche un fichier avec un diff binaire.....	2063
<u>8.140.9 xdiff_file_patch()</u> : Patche un fichier avec un diff unifié.....	2063
<u>8.140.10 xdiff_string_diff_binary()</u> : Créé un diff binaire de deux chaînes.....	2064
<u>8.140.11 xdiff_string_diff()</u> : Créé un diff unifié entre deux chaînes.....	2064
<u>8.140.12 xdiff_string_merge3()</u> : Fusionne trois chaînes en une seule.....	2065
<u>8.140.13 xdiff_string_patch_binary()</u> : Patche une chaîne avec un diff binaire.....	2065
<u>8.140.14 xdiff_string_patch()</u> : Patche une chaîne avec un diff unifié.....	2065
<u>8.141 Analyseur syntaxique XML</u>	2067
<u>8.141.1 Introduction</u>	2067
<u>8.141.2 Pré-requis</u>	2067
<u>8.141.3 Installation</u>	2067
<u>8.141.4 Configuration à l'exécution</u>	2067
<u>8.141.5 Types de ressources</u>	2068
<u>8.141.6 Constantes pré-définies</u>	2068

Sommaire

8 Référence des fonctions

<u>8.141.7 gestionnaires d'événements</u>	2068
<u>8.141.8 Problèmes de casse</u>	2069
<u>8.141.9 Codes d'erreurs</u>	2069
<u>8.141.10 Codage des caractères</u>	2070
<u>8.141.11 Exemples</u>	2070
<u>8.141.13 utf8_encode() : Convertit une chaîne ISO-8859-1 en UTF-8</u>	2075
<u>8.141.14 xml_error_string() : Lit le message d'erreur de l'analyseur XML</u>	2076
<u>8.141.15 xml_get_current_byte_index() : Retourne l'index de l'octet courant d'un analyseur XML</u>	2076
<u>8.141.16 xml_get_current_column_number() : Retourne le nombre courant de colonne d'un analyseur XML</u>	2076
<u>8.141.17 xml_get_current_line_number() : Retourne le numéro de ligne courant d'un analyseur XML</u>	2076
<u>8.141.18 xml_get_error_code() : Récupère le code erreur de l'analyseur XML</u>	2076
<u>8.141.19 xml_parse_into_struct() : Analyse une structure XML</u>	2076
<u>8.141.20 xml_parse() : Commence l'analyse d'un document XML</u>	2079
<u>8.141.21 xml_parser_create_ns() : Crée un analyseur XML</u>	2080
<u>8.141.22 xml_parser_create() : Création d'un analyseur XML</u>	2080
<u>8.141.23 xml_parser_free() : Détruit un analyseur XML</u>	2080
<u>8.141.24 xml_parser_get_option() : Lit les options d'un analyseur XML</u>	2081
<u>8.141.25 xml_parser_set_option() : Affecte les options d'un analyseur XML</u>	2081
<u>8.141.26 xml_set_character_data_handler() : Affecte les gestionnaires de texte littéral</u>	2081
<u>8.141.27 xml_set_default_handler() : Affecte le gestionnaire XML par défaut</u>	2082
<u>8.141.28 xml_set_element_handler() : Affecte les gestionnaires de début et de fin de balise XML</u>	2082
<u>8.141.29 xml_set_end_namespace_decl_handler() : Configure le gestionnaire XML de données</u>	2083
<u>8.141.30 xml_set_external_entity_ref_handler() : Configure le gestionnaire XML de références externes</u>	2084
<u>8.141.31 xml_set_notation_decl_handler() : Configure le gestionnaire XML de notations</u>	2084
<u>8.141.32 xml_set_object() : Utilise un analyseur XML à l'intérieur d'un objet</u>	2085
<u>8.141.33 xml_set_processing_instruction_handler() : Affecte les gestionnaires d'instructions exécutables</u>	2086
<u>8.141.34 xml_set_start_namespace_decl_handler() : Configure le gestionnaire de caractères</u>	2087
<u>8.141.35 xml_set_unparsed_entity_decl_handler() : Affecte les gestionnaires d'entités non déclarées</u>	2087
<u>8.142 XML-RPC</u>	2089
<u>8.142.1 Introduction</u>	2089
<u>8.142.2 Pré-requis</u>	2089
<u>8.142.3 Installation</u>	2089
<u>8.142.4 Configuration à l'exécution</u>	2089
<u>8.142.5 Types de ressources</u>	2089
<u>8.142.6 Constantes pré-définies</u>	2089
<u>8.142.8 xmlrpc_decode() : Décode le code XML en types PHP natifs</u>	2090
<u>8.142.9 xmlrpc_encode_request() : Génère le code XML d'une requête de méthode</u>	2090
<u>8.142.10 xmlrpc_encode() : Génère le code XML pour une valeur PHP</u>	2090
<u>8.142.11 xmlrpc_get_type() : Retourne le type XMLRPC d'une valeur PHP</u>	2091

Sommaire

8 Référence des fonctions

8.142.12 <u>xmlrpc_is_fault()</u> : Détermine si une valeur de tableau représente un XMLRPC.....	2091
8.142.13 <u>xmlrpc_parse_method_descriptions()</u> : Décode le code XML en une liste de descriptions de méthodes.....	2091
8.142.14 <u>xmlrpc_server_add_introspection_data()</u> : Ajoute des données d'introspection.....	2091
8.142.15 <u>xmlrpc_server_call_method()</u> : Analyse une requête XML est appelle les méthodes associées.....	2092
8.142.16 <u>xmlrpc_server_create()</u> : Crée un serveur XMLRPC.....	2092
8.142.17 <u>xmlrpc_server_destroy()</u> : Détruit un serveur XMLRPC.....	2092
8.142.18 <u>xmlrpc_server_register_introspection_callback()</u> : Enregistre une fonction PHP pour générer la documentation.....	2092
8.142.19 <u>xmlrpc_server_register_method()</u> : Enregistre une fonction PHP avec une méthode.....	2093
8.142.20 <u>xmlrpc_set_type()</u> : Modifie le type XMLRPC, base64 ou datetime, pour une chaîne de caractères PHP.....	2093
8.143 <u>XSL</u>	2094
8.143.1 <u>Introduction</u>	2094
8.143.2 <u>Pré-requis</u>	2094
8.143.3 <u>Installation</u>	2094
8.143.4 <u>Classes pré-définies</u>	2094
8.143.5 <u>Exemples</u>	2094
8.143.6 <u>Constantes pré-définies</u>	2095
8.143.8 <u>XSLTProcessor->getParameter()</u> : Récupère la valeur d'un paramètre.....	2095
8.143.9 <u>XSLTProcessor->hasExsltSupport()</u> : Indique si PHP utilise EXSLT.....	2096
8.143.10 <u>XSLTProcessor->importStylesheet()</u> : Importe une feuille de style.....	2096
8.143.11 <u>XSLTProcessor->registerPHPFunctions()</u> : Active l'utilisation de PHP dans les feuilles de styles XSLT.....	2096
8.143.12 <u>XSLTProcessor->removeParameter()</u> : Efface un paramètre.....	2096
8.143.13 <u>XSLTProcessor->setParameter()</u> : Définit la valeur d'un paramètre.....	2096
8.143.14 <u>XSLTProcessor->transformToDoc()</u> : Transforme en un document DOM.....	2096
8.143.15 <u>XSLTProcessor->transformToURI()</u> : Transforme en URI.....	2097
8.143.16 <u>XSLTProcessor->transformToXML()</u> : Transforme en XML.....	2097
8.144 <u>XSLT</u>	2098
8.144.1 <u>Introduction</u>	2098
8.144.2 <u>Pré-requis</u>	2098
8.144.3 <u>Installation</u>	2098
8.144.4 <u>Configuration à l'exécution</u>	2099
8.144.5 <u>Types de ressources</u>	2099
8.144.6 <u>Constantes pré-définies</u>	2099
8.144.8 <u>xslt_backend_name()</u> : Retourne le nom du backend.....	2100
8.144.9 <u>xslt_backend_version()</u> : Retourne le numéro de version de Sablotron.....	2100
8.144.10 <u>xslt_create()</u> : Crée un nouvel analyseur XSLT.....	2100
8.144.11 <u>xslt_erno()</u> : Retourne le numéro d'erreur XSLT courant.....	2100
8.144.12 <u>xslt_error()</u> : Retourne un message d'erreur.....	2100
8.144.13 <u>xslt_free()</u> : Détruit un analyseur XSLT.....	2101
8.144.14 <u>xslt_getopt()</u> : Récupère les options d'un processeur xsl donné.....	2101
8.144.15 <u>xslt_process()</u> : Transforme des données XML avec XSLT.....	2101
8.144.16 <u>xslt_set_base()</u> : Configure l'URI de base de toutes les transformations XSLT.....	2104

Sommaire

8 Référence des fonctions

8.144.17 <u>xsslt_set_encoding()</u> : Configure le jeu de caractères pour l'analyse des documents XML.....	2104
8.144.18 <u>xsslt_set_error_handler()</u> : Configure le gestionnaire d'erreurs du processeur XSLT.....	2104
8.144.19 <u>xsslt_set_log()</u> : Configure le fichier d'historique pour les messages XSLT.....	2104
8.144.20 <u>xsslt_set_object()</u> : Définit l'objet dans lequel doivent être résolues les fonctions de callback.....	2105
8.144.21 <u>xsslt_set_sax_handler()</u> : Modifie les gestionnaires SAX de l'analyseur XSLT.....	2105
8.144.22 <u>xsslt_set_sax_handlers()</u> : Configure les gestionnaires SAX qui seront appelés pour gérer les documents XML.....	2106
8.144.23 <u>xsslt_set_scheme_handler()</u> : Configure les gestionnaires de Scheme du processeur XSLT.....	2107
8.144.24 <u>xsslt_set_scheme_handlers()</u> : Configure un gestionnaire de Scheme pour un processeur XSLT.....	2107
8.144.25 <u>xsslt_setopt()</u> : Définit les options d'un processeur xsl donné.....	2107
8.145 <u>YAZ</u>	2108
8.145.1 <u>Introduction</u>	2108
8.145.2 <u>Installation</u>	2108
8.145.3 <u>Configuration à l'exécution</u>	2109
8.145.4 <u>Types de ressources</u>	2109
8.145.5 <u>Constantes pré-définies</u>	2109
8.145.6 <u>Exemples</u>	2109
8.145.8 <u>yaz_ccl_conf()</u> : Configure l'analyseur CCL.....	2111
8.145.9 <u>yaz_ccl_parse()</u> : Appelle l'analyseur CCL.....	2111
8.145.10 <u>yaz_close()</u> : Ferme une connexion YAZ.....	2111
8.145.11 <u>yaz_connect()</u> : Prépare une connexion à un serveur Z39.50.....	2111
8.145.12 <u>yaz_database()</u> : Spécifie la base d'une session YAZ.....	2112
8.145.13 <u>yaz_element()</u> : Spécifie le type d'éléments YAZ à lire.....	2112
8.145.14 <u>yaz_errno()</u> : Retourne le numéro d'erreur.....	2112
8.145.15 <u>yaz_error()</u> : Retourne une description de l'erreur.....	2112
8.145.16 <u>yaz_es_result()</u> : Inspecte le résultat Extended Services.....	2112
8.145.17 <u>yaz_get_option()</u> : Retourne la valeur d'une option de connexion YAZ.....	2113
8.145.18 <u>yaz_hits()</u> : Retourne le nombre de résultats de la dernière recherche.....	2113
8.145.19 <u>yaz_itemorder()</u> : Prépare une requête Z39.50 Item Order avec le package ILL-Request.....	2113
8.145.20 <u>yaz_present()</u> : Prépare à la lecture (Z39.50 present).....	2113
8.145.21 <u>yaz_range()</u> : Spécifie le nombre maximal de résultats à lire.....	2113
8.145.22 <u>yaz_record()</u> : Retourne un résultat.....	2113
8.145.23 <u>yaz_scan_result()</u> : Retourne le résultat d'un scan.....	2114
8.145.24 <u>yaz_scan()</u> : Prépare un scan.....	2114
8.145.25 <u>yaz_schema()</u> : Spécifie le schéma de lecture.....	2114
8.145.26 <u>yaz_search()</u> : Prépare une recherche.....	2114
8.145.27 <u>yaz_set_option()</u> : Modifie une ou plusieurs options de connexion YAZ.....	2114
8.145.28 <u>yaz_sort()</u> : Configure les critères de tri.....	2114
8.145.29 <u>yaz_syntax()</u> : Spécifie la syntaxe de lecture des lignes.....	2115
8.145.30 <u>yaz_wait()</u> : Attend l'exécution d'une requête.....	2115
8.146 <u>Accès aux fichiers Zip (Lecture seule)</u>	2116
8.146.1 <u>Introduction</u>	2116
8.146.2 <u>Pré-requis</u>	2116

Sommaire

8 Référence des fonctions

<u>8.146.3 Installation</u>	2116
<u>8.146.4 Configuration à l'exécution</u>	2116
<u>8.146.5 Types de ressources</u>	2116
<u>8.146.6 Constantes pré-définies</u>	2117
<u>8.146.7 Exemples</u>	2117
<u>8.146.9 zip_entry_close() : Ferme un dossier d'archive</u>	2118
<u>8.146.10 zip_entry_compressedsize() : Lit la taille compressée d'un dossier d'archives</u>	2118
<u>8.146.11 zip_entry_compressionmethod() : Lit la méthode de compression utilisée sur un dossier d'archives</u>	2118
<u>8.146.12 zip_entry_filesize() : Lit la taille décompressée d'un dossier d'archives</u>	2118
<u>8.146.13 zip_entry_name() : Lit le nom d'un dossier d'archives</u>	2118
<u>8.146.14 zip_entry_open() : Ouvre un dossier d'archives en lecture</u>	2118
<u>8.146.15 zip_entry_read() : Lit le contenu d'un fichier dans un dossier</u>	2118
<u>8.146.16 zip_open() : Ouvre une archive ZIP</u>	2119
<u>8.146.17 zip_read() : Lit la prochaine entrée dans une archive ZIP</u>	2119
8.147 Compression Zlib	2120
<u>8.147.1 Introduction</u>	2120
<u>8.147.2 Pré-requis</u>	2120
<u>8.147.3 Installation</u>	2120
<u>8.147.4 Configuration à l'exécution</u>	2120
<u>8.147.5 Types de ressources</u>	2121
<u>8.147.6 Constantes pré-définies</u>	2121
<u>8.147.7 Exemples</u>	2121
<u>8.147.9 gzcompress() : Comprime une chaîne</u>	2122
<u>8.147.10 gzdeflate() : Comprime une chaîne</u>	2123
<u>8.147.11 gzencode() : Crée une chaîne compressée gzip</u>	2123
<u>8.147.12 gzeof() : Indique si la fin d'un fichier compressé est atteinte</u>	2123
<u>8.147.13 gzfile() : Lit la totalité d'un fichier compressé</u>	2123
<u>8.147.14 gzgetc() : Lit un caractère dans un fichier compressé</u>	2123
<u>8.147.15 gzgets() : Lit une ligne dans un fichier compressé</u>	2124
<u>8.147.16 gzgetss() : Lit une ligne dans un fichier compressé, et supprime les balises HTML</u>	2124
<u>8.147.17 gzinflate() : Décompressé une chaîne</u>	2124
<u>8.147.18 gzopen() : Ouvre un fichier compressé avec gzip</u>	2124
<u>8.147.19 gzpassthru() : Affiche toutes les données qui restent dans un pointeur gz</u>	2124
<u>8.147.20 gzputs() : Alias de gzwrite</u>	2125
<u>8.147.21 gzread() : Lecture de fichier compressé binaire</u>	2125
<u>8.147.22 gzrewind() : Remplace le pointeur au début du fichier</u>	2125
<u>8.147.23 gzseek() : Déplace le pointeur de lecture</u>	2125
<u>8.147.24 gztell() : Lit la position courante du pointeur de lecture</u>	2125
<u>8.147.25 gzuncompress() : Décompressé une chaîne compressée</u>	2125
<u>8.147.26 gzwrite() : Écrit dans un fichier compressé gzip</u>	2125
<u>8.147.27 readgzfile() : Lit tout le fichier compressé</u>	2126
<u>8.147.28 zlib_get_coding_type() : Retourne la méthode de compression utilisée avec Gzip</u>	2126
<u>9 PHP et le moteur interne Zend</u>	2128
<u>9.1 API de flux : les auteurs d'extensions</u>	2128
<u>9.1.1 Présentation</u>	2128

Sommaire

9 PHP et le moteur interne Zend

<u>9.1.2</u>	<u>Eléments de bases</u>	2128
<u>9.1.3</u>	<u>Les flux qui sont des ressources</u>	2129
<u>9.2</u>	<u>internals.pdo</u>	2130
<u>9.3</u>	<u>API Zend : Modification du coeur de PHP</u>	2131
<u>9.4</u>	<u>Étendre PHP 3</u>	2131
<u>9.4.1</u>	<u>Créer une fonction PHP</u>	2131
<u>9.4.2</u>	<u>Appeler des fonctions utilisateurs</u>	2138
<u>9.4.3</u>	<u>Rapport d'erreurs</u>	2140

10 FAQ : foire Aux Questions.....2142

<u>10.1</u>	<u>Informations générales</u>	2142
<u>10.1.1</u>	<u>Qu'est-ce que PHP ?</u>	2142
<u>10.1.2</u>	<u>Que représente l'acronyme PHP ?</u>	2142
<u>10.1.3</u>	<u>Quelle est la relation entre les versions ?</u>	2142
<u>10.1.4</u>	<u>Puis-je utiliser plusieurs versions de PHP en même temps ?</u>	2142
<u>10.1.5</u>	<u>Quelle est la différence entre PHP 3 et PHP 4 ?</u>	2142
<u>10.1.6</u>	<u>Je pense avoir trouver un bogue ! Qui dois-je contacter ?</u>	2143
<u>10.2</u>	<u>Listes de diffusions</u>	2143
<u>10.2.1</u>	<u>Existe-t-il des listes de diffusions PHP ?</u>	2143
<u>10.2.2</u>	<u>Y a-t-il d'autre communauté ?</u>	2143
<u>10.2.3</u>	<u>Aidez-moi ! Je ne peux visiblement pas m'inscrire/désinscrire à une/d'une des listes de diffusions !</u>	2143
<u>10.2.4</u>	<u>Y a-t-il une archive des listes de diffusions quelque part ?</u>	2144
<u>10.2.5</u>	<u>Que puis-je demander sur les listes de diffusions ?</u>	2144
<u>10.2.6</u>	<u>Quelle information devrais-je inclure lors de l'écriture de mon message à la liste de diffusions ?</u>	2144
<u>10.3</u>	<u>Obtenir PHP</u>	2145
<u>10.3.1</u>	<u>Où puis-je obtenir PHP ?</u>	2145
<u>10.3.2</u>	<u>Est-ce que les versions binaires pré-compilées sont disponibles ?</u>	2145
<u>10.3.3</u>	<u>Où puis-je obtenir les bibliothèques dont j'ai besoin pour compiler les extensions optionnelles de PHP ?</u>	2145
<u>10.3.4</u>	<u>Comment faire fonctionner ces bibliothèques ?</u>	2146
<u>10.3.5</u>	<u>J'ai la dernière version du code source de PHP, téléchargé à partir de CVS. De quoi ai-je besoin pour le compiler sous Windows ?</u>	2146
<u>10.3.6</u>	<u>Où puis-je trouver le fichier Browser Capabilities ?</u>	2146
<u>10.4</u>	<u>Considérations sur les bases de données</u>	2146
<u>10.4.1</u>	<u>J'ai entendu dire qu'il était possible d'accéder à Microsoft SQL Server à partir de PHP. Comment est-ce possible?</u>	2146
<u>10.4.2</u>	<u>Puis-je accéder à des bases Microsoft Access</u>	2146
<u>10.4.3</u>	<u>J'ai mis à jour vers PHP 4, et maintenant MySQL me dit toujours "Warning: MySQL: Unable to save result set in ...". Que se passe-t-il?</u>	2148
<u>10.4.4</u>	<u>PHP 5 n'inclut plus les bibliothèques clientes MySQL, qu'est ce que ça implique pour moi? Puis-je toujours utiliser MySQL avec PHP? J'essaie d'utiliser MySQL, mais j'obtiens des erreurs "undefined function".....</u>	2148
<u>10.4.5</u>	<u>Après avoir ajouté le support partagé de MySQL, Apache plante dès que libphp4.so est chargé. Comment résoudre ce problème?</u>	2149
<u>10.4.6</u>	<u>Pourquoi est ce que j'obtiens une erreur comme celle-ci : "Warning: 0 is not a MySQL result index in <file> on line <x>" ou "Warning: Supplied argument is not a valid MySQL result resource in <file> on line <x>?".....</u>	2149
<u>10.5</u>	<u>FAQ sur l'installation</u>	2149
<u>10.5.1</u>	<u>Pourquoi ne doit-on pas utiliser Apache 2 dans un environnement threadé</u>	

Sommaire

10 FAQ : foire Aux Questions

<u>multi-processeur de production ?</u>	2149
<u>10.5.2 Unix/Windows : où doit être placé mon fichier</u>	2150
<u>10.5.3 Unix : j'ai installé PHP, mais à chaque fois que je charge un document, j'obtiens l'erreur 'Document Contains No Data'! Que se passe-t-il?</u>	2150
<u>10.5.4 Unix : J'ai installé PHP en utilisant des fichiers sources RPM, mais apache ne traite pas les pages PHP. Que se passe-t-il?</u>	2151
<u>10.5.5 Unix : J'ai installé PHP 3 en utilisant des paquets source RPM, mais ils ne compilent pas avec le support de la base de données dont j'ai besoin! Que se passe-t-il?</u>	2151
<u>10.5.6 Unix : J'ai patché Apache avec l'extension FrontPage et subitement, PHP ne fonctionne plus. Est-ce que PHP est incompatible avec l'extension FrontPage pour Apache ?</u>	2152
<u>10.5.7 Unix/Windows : J'ai installé PHP, mais lorsque je tente d'accéder à un fichier contenant un script PHP via mon navigateur, j'obtiens un écran vide</u>	2152
<u>10.5.8 Unix/Windows : J'ai installé PHP, mais lorsque je tente d'accéder à un fichier contenant un script PHP via mon navigateur, j'obtiens une erreur de type 'server 500 error'</u>	2153
<u>10.5.9 Quelques systèmes d'exploitations : J'ai installé PHP sans erreur, mais lorsque je tente de démarrer Apache, j'obtiens une erreur du type '</u>	2153
<u>10.5.10 Windows : J'ai installé PHP, mais lorsque j'accède à un fichier contenant un script PHP via mon navigateur, j'obtiens l'erreur :</u>	2153
<u>10.5.11 Windows : J'ai suivi toutes les instructions, mais je n'arrive toujours pas à faire fonctionner PHP et IIS ensemble !</u>	2153
<u>10.5.12 Lorsque vous exécutez PHP comme CGI avec IIS, PWS, OmniHTTPD ou Xitami, j'obtiens l'erreur suivante :</u>	2154
<u>10.5.13 Comment puis-je savoir si mon</u>	2154
<u>10.5.14 Où dois-je ajouter mon répertoire PHP à la variable</u>	2154
<u>10.5.15 Comment rendre le fichier</u>	2155
<u>10.5.16 Est-il possible d'utiliser la négociation sur le contenu fournie par Apache (option MultiViews) avec PHP ?</u>	2155
<u>10.5.17 Est-ce que PHP est limité au traitement des méthodes GET et POST ?</u>	2156
<u>10.6 Problèmes de compilation</u>	2156
<u>10.6.1 J'ai téléchargé la dernière version des sources de PHP en utilisant CVS, mais il n'y a pas de script configure!</u>	2156
<u>10.6.2 J'ai des problèmes pour configurer PHP avec Apache. On m'indique que</u>	2156
<u>10.6.3 Pendant la configuration de PHP (</u>	2156
<u>10.6.4 Quand je lance Apache, j'obtiens le message suivant :</u>	2156
<u>10.6.5 Quand je lance le ./configure, on me dit que les fichiers d'en-tête de GD, gdbm, ... ne sont pas trouvés!</u>	2157
<u>10.6.6 Quand le fichier</u>	2157
<u>10.6.7 Quand je lance</u>	2157
<u>10.6.8 Au moment de lier PHP, il y a des références indéfinies</u>	2157
<u>10.6.9 Je ne vois pas comment compiler PHP avec Apache 1.3</u>	2157
<u>10.6.10 j'ai suivi toutes les étapes pour installer le module Apache sous Unix, mais malgré tout, mes scripts PHP s'affichent en clair dans mon navigateur ou celui-ci me demande de sauver le fichier</u>	2158
<u>10.6.11 Il est dit d'utiliser</u>	2158
<u>10.6.12 Quand j'essaie de compiler Apache avec PHP en module statique en utilisant</u>	2158
<u>10.6.13 Quand j'essaie de compiler PHP avec</u>	2159
<u>10.6.14 Pendant le</u>	2159

Sommaire

10 FAQ : foire Aux Questions

<u>10.6.15 Quand je compile PHP avec le support MySQL, le configure se passe bien, mais pendant le.....</u>	2160
<u>10.6.16 Je veux mettre à jour mon PHP. Où puis-je trouver la ligne.....</u>	2160
<u>10.6.17 Quand je compile PHP avec le support de la bibliothèque GD, j'obtiens des erreurs de compilation étrange, voire même des erreurs de segmentation.....</u>	2160
<u>10.6.18 Quand je compile PHP, j'obtiens des erreurs aléatoires, voire même tout s'arrête. J'utilise Solaris.....</u>	2160
<u>10.7 Utiliser PHP.....</u>	2160
<u>10.7.1 J'aimerais écrire un script PHP générique qui pourrait traiter les données provenant de tout formulaire. Comment savoir quelles variables de la méthode POST sont disponibles ?.....</u>	2161
<u>10.7.2 Il faut que je convertisse tous les guillemets simples (') en un anti-slash suivi d'un guillemet simple (\'). Comment le faire avec une expression régulière ? J'aimerais aussi convertir " en \ " et \ en \\.....</u>	2161
<u>10.7.3 Tous mes " se transforment en \ " et mes ' en \' , comment me débarrasser de tous ces anti-slashes ? Comment et pourquoi sont-ils apparus ?.....</u>	2161
<u>10.7.4 Quand je fais ce qui suit, l'affichage se fait dans le mauvais ordre :.....</u>	2162
<u>10.7.5 Hey, où sont mes nouvelles lignes ?.....</u>	2162
<u>10.7.6 J'obtiens le message 'Warning: Cannot send session cookie - headers already sent...' ou 'Cannot add header information - headers already sent...'.....</u>	2162
<u>10.7.7 J'ai besoin d'accéder à des informations dans l'en-tête de requête directement. Comment puis-je le faire ?.....</u>	2162
<u>10.7.8 Quand j'essaye d'utiliser l'identification avec IIS j'obtiens.....</u>	2163
<u>10.7.9 Windows: Je ne peux pas accéder aux fichiers partagés sur un autre ordinateur utilisant IIS.....</u>	2163
<u>10.7.10 Mon script PHP fonctionne avec IE et Lynx, mais avec Netscape une portion de ce qui devrait s'afficher manque. Quand j'affiche la source HTML de la page, je vois le contenu avec IE mais pas avec Netscape.....</u>	2163
<u>10.7.11 Comment mélanger XML et PHP ? PHP se plaint de mes balises <?xml !.....</u>	2163
<u>10.7.12 Comment utiliser PHP avec FrontPage ou d'autres éditeurs HTML qui insistent pour mettre mon code ailleurs ?.....</u>	2164
<u>10.7.13 Où puis-je trouver une liste complète des variables pré-définies que je peux utiliser dans mes scripts PHP ?.....</u>	2164
<u>10.7.14 Comment puis-je générer des fichiers PDF sans utiliser les bibliothèques non libres.....</u>	2164
<u>10.7.15 J'essaye d'accéder à une des variables standard CGI (comme.....</u>	2164
<u>10.7.16 Quelques directives PHP peuvent également prendre des valeurs d'octets sténographiées contrairement aux valeurs d'octets uniquement.....</u>	2165
<u>10.8 PHP et HTML.....</u>	2165
<u>10.8.1 Quel encodage/décodage ai-je besoin lors du passage d'une valeur via un formulaire/une URL ?.....</u>	2165
<u>10.8.2 J'essaye d'utiliser <input type="image"> mais les variables.....</u>	2166
<u>10.8.3 Comment créer un tableau dans une balise <form> HTML ?.....</u>	2166
<u>10.8.4 Comment puis-je récupérer le résultat d'un champs HTML.....</u>	2167
<u>10.8.5 Comment puis-je passer une variable de Javascript vers PHP ?.....</u>	2168
<u>10.9 PHP et COM.....</u>	2168
<u>10.9.1 J'ai compilé une DLL pour calculer quelquechose. Y a-t-il un moyen d'utiliser cette DLL sous PHP?.....</u>	2168
<u>10.9.2 Que signifie 'Unsupported variant type: xxxx (0xxxxx)'?.....</u>	2168
<u>10.9.3 Est-il possible de manipuler des objets visuels en PHP?.....</u>	2169

Sommaire

10 FAQ : foire Aux Questions

<u>10.9.4 Puis-je stocker un objet COM dans une session?</u>	2169
<u>10.9.5 Comment puis-je intercepter des erreurs COM?</u>	2169
<u>10.9.6 Puis-je générer des fichiers DLL à partir de PHP comme je le fais avec Perl?</u>	2169
<u>10.9.7 Que signifie 'Unable to obtain IDispatch interface for CLSID {xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}'?</u>	2169
<u>10.9.8 Comment puis-je utiliser des objets COM à partir d'un serveur distant?</u>	2169
<u>10.9.9 J'obtiens 'DCOM is disabled in C:\path...\scriptname.php on line 6', que dois-je faire?</u>	2169
<u>10.9.10 Est-il possible de charger/manipuler des objets ActiveX dans une page en PHP ?</u>	2170
<u>10.9.11 Est-il possible d'obtenir une instance fonctionnelle d'un composant?</u>	2170
<u>10.9.12 Y a-t-il moyen de manipuler un événement envoyé par un objet COM?</u>	2170
<u>10.9.13 J'ai des problèmes quand j'invoque une méthode d'un objet COM qui expose plus d'une interface. Que puis-je faire?</u>	2170
<u>10.9.14 Bon, PHP fonctionne avec COM, mais qu'en est-il de COM+?</u>	2170
<u>10.9.15 Si PHP peut manipuler des objets COM, peut-on imaginer d'utiliser des ressources de composants, en conjonction avec PHP?</u>	2170
<u>10.10 PHP et les autres langages</u>	2171
<u>10.10.1 PHP contre ASP?</u>	2171
<u>10.10.2 Existe-t-il un convertisseur de ASP vers PHP?</u>	2171
<u>10.10.3 PHP contre Cold Fusion?</u>	2171
<u>10.10.4 PHP contre Perl?</u>	2172
<u>10.11 Migrer de PHP 2 à PHP 3</u>	2172
<u>10.11.1 Migrer de PHP 2 à PHP 3 ?</u>	2172
<u>10.12 Migrer de PHP 3 à PHP 4</u>	2172
<u>10.12.1 Migrer de PHP 3 à PHP 4</u>	2172
<u>10.12.2 Les sessions fonctionnent-elles avec PHP 3 ?</u>	2172
<u>10.12.3 Fonctions incompatibles ?</u>	2172
<u>10.13 Migration de PHP 4 à PHP 5</u>	2173
<u>10.13.1 Migration de PHP 4 à PHP 5</u>	2173
<u>10.13.2 Est-ce que MySQL fonctionne en PHP 5 ? Il semble avoir disparu</u>	2173
<u>10.13.3 J'ai entendu dire que PHP 5 à un tout nouveau model objet : est-ce que mon actuel code objet fonctionne ? Où puis-je trouver des informations concernant ces nouvelles fonctionnalités ?</u>	2173
<u>10.13.4 Donc, mise à part le nouveau model objet, qu'est-ce qui a changé dans PHP 5 ? En particulier, existe-t-il une version spécifique du manuel PHP ?</u>	2173
<u>10.14 Questions diverses</u>	2174
<u>10.14.1 Comment puis-je manipuler le manuel compressé en bzip2 sous Windows ?</u>	2174
<u>10.14.2 Que signifie le caractère "&" devant les arguments dans les déclarations des fonctions comme la fonction</u>	2174
<u>10.14.3 Comment puis-je gérer la directive</u>	2174

11 Annexes.....**2176**

<u>11.1 Histoire de PHP</u>	2176
<u>11.1.1 Histoire de PHP</u>	2176
<u>11.1.2 Quelques projets liés à PHP</u>	2177
<u>11.1.3 Livres traitant de PHP</u>	2179
<u>11.1.4 Publications sur PHP</u>	2179
<u>11.2 Migration de PHP 4 vers PHP 5</u>	2179

Sommaire

11 Annexes

11.2.1 Nouveautés de PHP 5.....	2179
11.2.2 Incompatibilités avec les version antérieures.....	2179
11.2.3 CLI et CGI.....	2181
11.2.4 Migration des fichiers de configuration.....	2181
11.2.5 Nouvelles fonctions.....	2182
11.2.6 Nouvelles Directives.....	2184
11.2.7 Bases de données.....	2185
11.2.8 Nouveau Modèle Objet.....	2185
11.2.9 Rapport d'erreurs.....	2185
11.3 Migration de PHP 3 à PHP 4.....	2186
11.3.1 Ce qui a changé en PHP 4.....	2186
11.3.2 Utiliser PHP 3 et PHP 4 simultanément.....	2186
11.3.3 Migration des fichiers de configuration.....	2186
11.3.4 Comportement de l'analyseur.....	2187
11.3.5 Rapport d'erreur.....	2188
11.3.6 Initialiseur.....	2189
11.3.7 empty("0").....	2189
11.3.8 Fonctions manquantes.....	2189
11.3.9 Extensions PHP 3.....	2190
11.3.10 Substitution de variables dans les chaînes.....	2190
11.3.11 Cookies.....	2191
11.3.12 Gestion des variables globales.....	2191
11.4 Migration de PHP/FI 2 à PHP 3.....	2191
11.4.1 A propos des incompatibilités en 3.0.....	2191
11.4.2 old_function.....	2192
11.4.3 Balises PHP.....	2192
11.4.4 Syntaxe if...endif.....	2192
11.4.5 Syntaxe while.....	2193
11.4.6 Types d'expression.....	2193
11.4.7 Les messages d'erreur ont changé.....	2194
11.4.8 Evaluation rapide des booléens.....	2194
11.4.9 La valeur TRUE / FALSE comme retour de fonctions.....	2194
11.4.10 Diverses incompatibilités.....	2195
11.5 Débogueur PHP.....	2196
11.5.1 A propos du débogueur.....	2196
11.5.2 Utiliser le débogueur PHP.....	2196
11.5.3 Protocole du débogueur.....	2196
11.6 Options de configuration.....	2197
11.6.1 Listes des options de configuration internes.....	2197
11.7 Directives du php.ini.....	2201
11.7.1 Liste des directives du php.ini.....	2201
11.7.2 Description des directives internes du php.ini.....	2214
11.8 Liste des Fuseaux Horaires Supportés.....	2221
11.8.1 Liste des fuseaux horaires dans le groupe Africa (Afrique).....	2221
11.8.2 Liste des fuseaux horaires dans le groupe America (Amérique).....	2222
11.8.3 List of timezones in the group Antarctica.....	2222
11.8.4 List of timezones in the group Arctic.....	2223
11.8.5 Liste des fuseaux horaires dans le groupe Asia (Asie).....	2223
11.8.6 Liste des fuseaux horaires dans le groupe Atlantic (Atlantique).....	2223
11.8.7 List of timezones in the group Australia.....	2223
11.8.8 Liste des fuseaux horaires dans le groupe Europe.....	2223

Sommaire

11 Annexes

<u>11.8.9 Liste des fuseaux horaires dans le groupe Pacific (Pacifique)</u>	2224
<u>11.8.10 Liste des fuseaux horaires dans le groupe Autres</u>	2224
<u>11.9 extensions</u>	2224
<u>extensions.membership</u>	2224
<u>extensions.state</u>	2225
<u>11.10 Liste des alias</u>	2225
<u>11.11 Mots réservés en PHP</u>	2232
<u>11.11.1 Liste de mots-clé</u>	2232
<u>11.11.2 Variables pré-définies</u>	2232
<u>11.11.3 Classes pré-définies</u>	2239
<u>11.11.4 Constantes pré-définies</u>	2239
<u>11.12 Types des ressources PHP</u>	2244
<u>11.13 Liste des protocoles supportés</u>	2257
<u>11.13.1 Système de fichiers</u>	2258
<u>11.13.2 Socket</u>	2258
<u>11.13.3 HTTP et HTTPS</u>	2259
<u>11.13.4 FTP et FTPS</u>	2261
<u>11.13.5 Flux d'entrée/sortie</u>	2262
<u>11.13.6 Flux compressés</u>	2263
<u>11.13.7 Shell sécurisé 2</u>	2264
<u>11.13.8 Flux Audio</u>	2265
<u>11.13.9 Gestionnaire de flux</u>	2267
<u>11.14 Liste des filtres standards</u>	2267
<u>11.14.1 Filtres de chaînes de caractères</u>	2267
<u>11.14.2 Filtres de conversion</u>	2269
<u>11.14.3 Filtres de compression</u>	2269
<u>11.14.4 Filtres de chiffrement</u>	2272
<u>11.15 Liste des modes de transports de sockets disponibles</u>	2273
<u>11.15.1 Domaines Internet : TCP, UDP, SSL et TLS</u>	2273
<u>11.15.2 Domaine Unix : UNIX et UDG</u>	2274
<u>11.16 Comparaison de types en PHP</u>	2274
<u>11.17 Liste des token de l'analyseur</u>	2276
<u>11.18 A propos du manuel</u>	2279
<u>11.18.1 Formats</u>	2279
<u>11.18.2 A propos des notes utilisateurs</u>	2280
<u>11.18.3 Comment lire la définition d'une fonction (prototype)</u>	2280
<u>11.18.4 Versions de PHP documentées dans ce manuel</u>	2281
<u>11.18.5 Où trouver plus d'informations sur PHP ?</u>	2282
<u>11.18.6 Comment aider à l'amélioration de la documentation</u>	2282
<u>11.18.7 Comment sont générées les documentations</u>	2283
<u>11.18.8 Traductions</u>	2283
<u>11.18.9 Traduction française</u>	2283
<u>11.19 Open Publication License</u>	2284
<u>11.19.1 I. REQUIREMENTS ON BOTH UNMODIFIED AND MODIFIED VERSIONS</u>	2284
<u>11.19.2 II. COPYRIGHT</u>	2284
<u>11.19.3 III. SCOPE OF LICENSE</u>	2284
<u>11.19.4 IV. REQUIREMENTS ON MODIFIED WORKS</u>	2285
<u>11.19.5 V. GOOD-PRACTICE RECOMMENDATIONS</u>	2285
<u>11.19.6 VI. LICENSE OPTIONS</u>	2285

Sommaire

<u>12 Version HTML Help</u>	2287
<u>13 Index</u>	2288
<u>13.1 Index des fonctions</u>	2288
<u>13.2 Index des concepts</u>	2364
<u>13.3 Index des exemples</u>	2443

Manuel PHP

Date de génération : 04/12/2005 09:12:14

1.1 translators

- Avec, pour la traduction (par ordre alphabétique) :
- Mehdi Achour
- Vincent Briet
- Jean-Sébastien Goupil
- David Manusset
- Guillaume Plessis
- Damien Seguy
- Yannick Torres

© 1997-2005, *PHP Documentation Group* ,

1.2 Note légale

Copyright

Copyright © 1997 - 2005 par le PHP Documentation Group. Ce document ne peut être redistribué qu'aux termes et aux conditions mentionnées dans la licence Open Publication License , version 1.0 ou plus récente. Une copie de l' Open Publication License est distribuée avec ce manuel; la dernière version est disponible à <http://www.opencontent.org/openpub/> .

La distribution de versions adaptées de ce document est interdite sans la permission explicite des propriétaires du copyright.

La distribution de ce travail ou d'un dérivé de celui-ci sous n'importe quel forme (papier) est interdite sans l'accord des propriétaires du copyright.

Dans le cas où vous seriez intéressés par la distribution partielle ou totale de ce document (modifié ou non) et que vous ayez des questions, merci de contacter les propriétaires du copyright à doc-license@lists.php.net . Notez que cette adresse courriel est directement redirigée vers une liste de diffusion publique et archivée.

La section API Zend : Modification du coeur de PHP de la documentation est basée sur une contribution initiale de Zend Technologies.

2 Préface

Introduction

Ce manuel est essentiellement une référence des fonctions , mais il contient aussi des informations de référence sur le langage , des explications sur les fonctionnalités principales de PHP et diverses informations supplémentaires .

Vous pouvez télécharger ce manuel sous divers formats, sur <http://www.php.net/download-docs.php> , et sur le site de <http://www.nexen.net/> . Plus d'informations sur ce manuel sont disponibles dans l'appendice A propos du manuel . Si vous voulez découvrir l'histoire de PHP .

2.1 Auteurs et Contributeurs

Nous mettons en avant les personnes les plus actives dans la préface du manuel mais il y a bien plus de contributeurs qui nous aident actuellement dans notre travail ou qui ont fournis une aide précieuse au projet dans le passé. Il y a beaucoup d'inconnus qui nous ont aidés à travers leurs notes concernant les pages du manuel qui sont continuellement incluses dans le manuel, travail dont nous sommes très reconnaissants. La liste fournie ci-dessous est classée par ordre alphabétique.

Auteurs et Editeurs

Les contributeurs suivants ont eu un impact énorme en ajoutant du contenu dans le manuel : Jouni Ahto, Alexander Aulbach, Daniel Beckham, Stig Bakken, Jesus M. Castagnetto, Ron Chmara, Sean Coates, John Coggeshall, Simone Cortesi, Markus Fischer, Wez Furlong, Sara Golemon, Rui Hirokawa, Brad House, Moriyoshi Koizumi, Rasmus Lerdorf, Andrew Lindeman, Stanislav Malyshev, Rafael Martinez, Yasuo Ohgaki, Derick Rethans, Sander Roobol, Egon Schmid, Thomas Schoefbeck, Sascha Schumann, Dan Scott, Lars Torben Wilson, Jim Winstead, Jeroen van Wolffelaar et Andrei Zmievski.

Les contributeurs suivants ont énormément aidé dans l'édition du manuel : Stig Bakken, Hartmut Holzgraefe et Egon Schmid.

Mainteneurs des notes utilisateurs

Les mainteneurs actuellement les plus actifs sont : Mehdi Achour, Friedhelm Betz, Vincent Gevers, Aidan Lister, Nuno Lopes et Tom Sommer.

Ces personnes ont également déployé énormément d'efforts dans le maintien des notes utilisateurs : Daniel Beckham, Victor Boivie, Jesus M. Castagnetto, Nicolas Chaillan, Ron Chmara, James Cox, Sara Golemon, Zak Greant, Szabolcs Heilig, Oliver Hinckel, Hartmut Holzgraefe, Rasmus Lerdorf, Andrew Lindeman, Maxim Maletsky, James Moore, Sebastian Picklum, Derick Rethans, Sander Roobol, Damien Seguy, Jason Sheets, Jani Taskinen, Yasuo Ohgaki, Philip Olson, Lars Torben Wilson, Jim Winstead, Jared Wyles et Jeroen van Wolffelaar.

3 Au moment de commencer

3.1 Introduction

3.1.1 Qu'est ce que PHP?

PHP (officiellement, ce sigle est un acronyme récursif pour PHP: Hypertext Preprocessor) est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web. Il peut être intégré facilement au HTML.

Réponse simple et claire, mais qu'est ce que cela veut dire? Un exemple :

Exemple d'introduction
<pre><html> <head> <title>Exemple</title> </head> <body> <?php echo "Bonjour, je suis un script PHP!"; ?> </body> </html></pre>

Il est à noter la différence avec les autres scripts CGI écrits dans d'autres langages tels que le Perl ou le C : Au lieu d'écrire un programme avec de nombreuses lignes de commandes afin d'afficher une page HTML, vous écrivez une page HTML avec du code inclus à l'intérieur afin de réaliser une action précise (dans ce cas là, afficher du texte). Le code PHP est inclus entre une balise de début et une balise de fin qui permettent au serveur web de passer en "mode PHP".

Ce qui distingue PHP des langages de script comme le Javascript est que le code est exécuté sur le serveur. Si vous avez un script similaire sur votre serveur, le client ne reçoit que le résultat du script, sans aucun moyen d'avoir accès au code qui a produit ce résultat. Vous pouvez configurer votre serveur web afin qu'il analyse tous vos fichiers HTML comme des fichiers PHP. Ainsi, il n'y a aucun moyen de distinguer les pages qui sont produites dynamiquement des pages statiques.

Le grand avantage de PHP est qu'il est extrêmement simple pour les néophytes, mais offre des fonctionnalités avancées pour les experts. Ne craignez pas de lire la longue liste de fonctionnalités PHP. Vous pouvez vous plonger dans le code, et en quelques instants, écrire des scripts simples.

Bien que le développement de PHP soit orienté vers la programmation pour les sites web, vous pouvez en faire bien d'autres usages. Lisez donc la section Que peut faire PHP? ou bien le tutoriel d'introduction si vous êtes uniquement intéressé dans la programmation web.

3.1.2 Que peut faire PHP?

Tout. PHP est principalement conçu pour servir de langage de script coté serveur, ce qui fait qu'il est capable de réaliser tout ce qu'un script CGI quelconque peut faire, comme collecter des données de formulaire, générer du contenu dynamique, ou gérer des cookies. Mais PHP peut en faire bien plus.

Il y a trois domaines différents où PHP peut s'illustrer.

- Langage de script coté serveur. C'est l'utilisation la plus traditionnelle, et aussi le principal objet de PHP. Vous aurez besoin de trois composants pour l'exploiter : un analyseur PHP (CGI ou module serveur), un serveur web et un navigateur web. Vous devez exécuter le serveur web en corrélation avec PHP. Vous pouvez accéder au programme PHP avec l'aide du navigateur web. Tout ceci peut fonctionner sur votre propre machine si vous êtes juste expérimenté avec la programmation en PHP. Voyez la section [d'installation](#) pour plus d'informations.
- Langage de programmation en ligne de commande. Vous pouvez écrire des scripts PHP et l'exécuter en ligne de commande, sans l'aide du serveur web et d'un navigateur. Il vous suffit de disposer de l'exécutable PHP. Cette utilisation est idéale pour les scripts qui sont exécutés régulièrement (avec un cron sous Unix ou Linux), ou un Task Scheduler (sous Windows). Ces scripts peuvent aussi être utilisés pour réaliser des opérations sur des fichiers texte. Voyez la section sur l'utilisation de PHP en [ligne de commande](#) pour plus d'informations.
- Ecrire des applications clientes graphiques. PHP n'est probablement pas le meilleur langage pour écrire des applications clientes graphiques, mais si vous connaissez bien PHP et que vous souhaitez exploiter des fonctionnalités avancées dans vos applications clientes, vous pouvez utiliser PHP-GTK pour écrire de tels programmes. Vous avez aussi la possibilité d'écrire des applications très portables avec ce langage. PHP-GTK est une extension de PHP, qui n'est pas fournie dans la distribution de base. Si vous êtes intéressé par PHP-GTK, visitez [son site web](#) .

PHP est utilisable sur la majorité des systèmes d'exploitation, comme Linux, de nombreuses variantes Unix (incluant HP-UX, Solaris et OpenBSD), Microsoft Windows, Mac OS X, RISC OS et d'autres encore. PHP supporte aussi la plupart des serveurs web actuels : Apache, Microsoft Internet Information Server, Personal Web Server, Netscape et iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd et beaucoup d'autres encore. Pour la majorité des serveurs web, PHP fonctionne comme module, et pour d'autres, il fonctionne comme exécutable CGI.

Avec PHP vous avez le choix de votre système d'exploitation et de votre serveur web. De plus, vous avez aussi le choix d'utiliser la programmation procédurale ou objet, ou encore un mélange des deux. Bien que le support de la couche objet ne soit pas très standard en PHP 4, beaucoup de bibliothèques et d'applications d'envergures (incluant la bibliothèque PEAR) ont été écrites en utilisant uniquement du code orienté objet. PHP 5 a rectifié les faiblesses de la couche objet de PHP 4 et a introduit un modèle objet complet.

Avec PHP, vous n'êtes pas limités à la production de code HTML. Les capacités de PHP lui permettent de générer aussi bien des images, des fichiers PDF, des animations Flash (avec l'aide des bibliothèques libswf et Ming), générés à la volée. Vous pouvez aussi générer facilement du texte, du code XML ou XHTML. PHP génère tous ces fichiers, et les sauve dans le système de fichier, ou bien les envoie directement au navigateur web.

Une des grandes forces de PHP est le support de nombreuses bases de données. Ecrire une page web exploitant une base de données est extrêmement simple. Les bases de données suivantes sont toutes supportées par PHP :

- Adabas D
- dBase
- Empress
- FilePro (lecture seule)
- Hyperwave

- IBM DB2
- Informix
- Ingres
- InterBase
- FrontBase
- mSQL
- Direct MS-SQL
- MySQL
- ODBC
- Oracle (OCI7 et OCI8)
- Ovrimos
- PostgreSQL
- SQLite
- Solid
- Sybase
- Velocis
- Unix dbm

Il existe aussi des couches d'abstraction de base de données comme DBX qui vous permettent de vous connecter de manière transparente à toute base de données supportée par cette extension. De plus, PHP supporte ODBC, ce qui vous permet de vous connecter à toute autre base de données qui supporte ce standard.

PHP supporte de nombreux protocoles comme LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (sous Windows) et encore d'autres. Vous pouvez ouvrir des sockets réseau, et interagir avec n'importe quel autre protocole. PHP supporte le format complexe WDDX, qui permet de communiquer entre tous les langages web. En terme d'interconnexion, PHP supporte aussi les instanciations d'objets Java, et les utilise de manière transparente comme objets intégrés. Vous pouvez aussi exploiter les objets distants avec CORBA.

PHP dispose de fonctionnalités extrêmement utiles pour le traitement de texte, allant des expressions rationnelles POSIX étendues ou Perl aux traitements des fichiers XML, avec les standards SAX et DOM (PHP 4). Vous pouvez utiliser les transformations XSLT. PHP 5 standardise toutes les extensions XML sur la base solide de libxml2 et en attendant les fonctionnalités en ajoutant le support de SimpleXML et XMLReader.

En commerce électronique, vous trouverez des outils de paiement intégrés comme Cybercash, CyberMut, VeriSign Payflow Pro et MCVE, pour réaliser des paiements en ligne.

Enfin, PHP dispose d'extensions très pratiques comme le moteur de recherche mnoGoSearch, la passerelle avec IRC, des outils de compression (gzip, bz2) et de conversion calendaire, de traduction...

Comme vous le voyez, cette page n'est pas assez grande pour lister toutes les puissantes fonctionnalités de PHP. Lisez la section sur [l'installation de PHP](#) et étudiez la [liste de fonctions](#) pour avoir plus de détails sur toutes ces technologies.

3.2 Une introduction à PHP

Dans cette section, nous voulons illustrer les principes de base de PHP dans une courte introduction. Ce chapitre traite uniquement de créations de pages web dynamiques avec PHP, laissant de côté temporairement les autres possibilités de PHP. Voyez la section [Ce que peut faire PHP](#) pour plus d'informations.

Les pages web qui exploitent PHP sont traitées comme des pages HTML standards, et vous pouvez les créer, éditer et effacer tout comme vous le faites normalement avec des pages HTML classiques.

3.2.1 Le nécessaire

Dans ce tutoriel, nous présumons que vous avez un serveur web avec le support PHP activé, et que les fichiers terminés par l'extension .php sont traités par PHP. Sur la plupart des serveurs, c'est la configuration par défaut, mais n'hésitez pas à interroger votre administrateur système en cas de doute. Si votre serveur web supporte PHP, vous n'avez rien à faire. Simplement, créez un dossier, puis créez un fichier texte, avec l'extension .php : le serveur va automatiquement l'exécuter avec PHP. Il n'y a pas de compilation, ou d'installation compliquée. Gardez en tête que les fichiers sont comparables à des fichiers HTML, dans lesquels vous allez utiliser des balises magiques, qui feront beaucoup de choses pour vous. Beaucoup d'hébergeurs web proposent PHP mais si votre hébergeur ne le propose pas, lisez la section des [liens PHP](#) pour trouver un hébergeur web le proposant.

Supposons que vous souhaitez économiser du temps en ligne et travailler localement. Dans ce cas, vous devez installer un serveur web comme [Apache](#) , et bien sur [PHP](#) . Vous souhaitez aussi installer une base de données comme [MySQL](#) .

Vous pouvez soit installer ces logiciels individuellement ou bien d'une manière simplifiée. Notre manuel contient les [instructions d'installation de PHP](#) (en supposant que vous avez déjà un serveur web d'installer). Dans le cas où vous rencontrez des problèmes dans l'installation de PHP, nous vous suggérons de poser vos questions sur notre [liste de diffusions](#) réservée à cette usage. Si vous choisissez une version simplifiée, vous pouvez utiliser des [des installeurs](#) qui prennent en charge l'ensemble de l'installation en quelques clics. Il est facile de configurer un serveur web avec le support de PHP sur n'importe quel système d'exploitation, y compris MacOS, Linux et Windows. Sous Linux, vous pouvez aussi trouver des commandes comme [rpmfind](#) et [PBone](#) très pratique pour rechercher les paquets pré-compilés. Vous pouvez aussi visiter [apt-get](#) , pour des paquets Debian.

3.2.2 Votre première page PHP

Créez un fichier appelé `bonjour.php` dans votre dossier web racine (`DOCUMENT_ROOT`) avec le contenu suivant :

Notre premier script PHP : `bonjour.php`

```
<html>
<head>
  <title>Test PHP</title>
</head>
<body>
  <?php echo '<p>Bonjour le monde</p>'; ?>
</body>
</html>
```

Utilisez votre navigateur pour accéder au fichier via votre serveur web, en ajoutant le nom de fichier `/bonjour.php` . Si vous développez localement, votre URL ressemblera à `http://localhost/bonjour.php` ou encore `http://127.0.0.1/bonjour.php` mais cela dépend de la configuration de votre serveur web. Si ceci est configuré correctement, le fichier sera analysé par PHP et le résultat suivant sera affiché :

```

<html>
<head>
<title>Test PHP</title>
</head>
<body>
<p>Bonjour le monde</p>
</body>
</html>

```

Ce programme est extrêmement simple et vous n'avez pas besoin de PHP pour créer une page web comme ceci. Elle ne fait qu'afficher Bonjour le monde , grâce à la fonction `echo` de PHP. Notez que ce fichier **n'a pas besoin d'être exécutables** ou autres, dans aucun cas. Le serveur sait que ce fichier a besoin d'être interprété par PHP car vous utilisez l'extension ".php", et le serveur est configuré pour les passer à PHP. Voyez cela comme une page HTML normale qui contient une série de balises spéciales qui va vous permettre de réaliser beaucoup de choses intéressantes.

Si vous avez essayé cet exemple, et qu'il n'a rien affiché de spécial, ou même qu'une boîte de dialogue a surgi pour vous proposer de le télécharger, ou encore vous avez vu le code tel que nous l'avons écrit dans le fichier, alors votre serveur web ne supporte probablement pas PHP ou est mal configuré. Demandez à votre administrateur de l'activer pour vous, en utilisant le chapitre [Installation](#) . Si vous développer localement, lisez également le chapitre d'installation afin de vous assurer que tout est configuré correctement. Assurez-vous que vous tentez d'accéder au fichier via http et que le serveur web vous fournisse la sortie. Si vous appelez votre fichier depuis votre gestionnaire de fichiers, il ne sera pas analysé par PHP. Si le problème persiste malgré cela, n'hésitez pas à utiliser une [des options de support](#) de PHP.

Le point important de cet exemple était de montrer le format des balises spéciales PHP. Nous avons utilisé ici `<?php` pour indiquer le début de la balise PHP. Puis, nous avons introduit les commandes PHP et refermé les balises PHP avec `?>` . Vous pouvez passer du mode PHP au mode HTML et vice-versa, de cette manière, et à votre guise. Pour plus d'informations, lisez la section du manuel sur la [syntaxe basique de PHP](#) .

Note

Une note sur les éditeurs de texte

Il existe de nombreux éditeurs de texte et environnements de développement (IDE) que vous pouvez utiliser pour créer, éditer et gérer vos applications PHP. Une liste partielle de ces outils est entretenue à l'adresse [PHP Editor's List](#) . Si vous voulez recommander un éditeur particulier, rendez donc une visite à cette page, et demandez au webmestre d'ajouter votre éditeur. Avoir au minimum un éditeur de texte avec la coloration syntaxique est vivement recommandé.

Note

Une note sur les traitements de texte

Les traitements de texte tels que StarOffice Writer, Microsoft Word et Abiword sont de très mauvais choix pour éditer des scripts PHP. Si vous voulez utiliser l'un d'entre eux, malgré tout, pour tester vos scripts, vous devez vous assurer que vous sauvez les fichiers au format texte seul (**plain text**) : sinon, PHP ne sera pas capable de lire et d'exécuter ces scripts.

Note

Une note sur le Notepad de Windows

Si vous écrivez vos scripts PHP avec Windows Notepad, vous devez vous assurer que vos fichiers sont sauvés avec l'extension .php (Notepad ajoute automatiquement une extension .txt à vos fichiers, à moins que vous ne preniez l'une des mesures suivantes). Lorsque vous sauvez un

fichier, et que vous êtes invité à lui donner un nom, placez le nom du fichier entre doubles guillemets (i.e. " hello.php "). Vous pouvez également cliquer dans le menu 'Documents texte' du dialogue de sauvegarde, et choisir l'option 'Tous les fichiers'. Vous pourrez alors saisir le nom de votre fichier sans les doubles guillemets.

Maintenant vous avez créé un script PHP fonctionnant, c'est le moment de créer le meilleur script PHP ! Faites un appel à la fonction `phpinfo` et vous verrez beaucoup d'informations intéressantes sur votre système et sa configuration comme les variables pré-définies disponibles , les modules PHP chargés ainsi que la configuration . Prenez du temps pour revoir ces informations importantes.

Récupération des informations du système depuis PHP

```
<?php phpinfo(); ?>
```

3.2.3 Trucs pratiques

Réalisons maintenant quelque chose de plus puissant. Nous allons vérifier le type de navigateur que le visiteur de notre site utilise. Pour cela, nous allons accéder aux informations que le navigateur du visiteur nous envoie, lors de sa requête HTTP. Cette information est stockée dans une variable . Les variables sont faciles à repérer, car elles commencent toutes par un signe dollar. La variable qui nous intéresse ici est `$_SERVER['HTTP_USER_AGENT']` .

Note

`$_SERVER` est une variable spéciale de PHP, qui contient toutes les informations relatives au serveur web. C'est une variable réservée de PHP, et une auto-globale. Reportez-vous aux pages du manuel traitant des Auto-globales (aussi connues sous le nom de super-globales). Ces variables spéciales ont été introduites en PHP 4.1.0 . Auparavant, il fallait utiliser les variables `$HTTP_*_VARS` , comme `$HTTP_SERVER_VARS` . Bien qu'obsolètes, ces variables existent toujours. (Voir aussi la note sur l'ancien code .)

Pour afficher cette variable, nous pouvons simplement faire :

Afficher le contenu d'une variable (élément de tableau)

```
<?php echo $_SERVER['HTTP_USER_AGENT']; ?>
```

Un résultat possible du script pourra alors être :

Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)

Il y a de nombreux types de variables disponibles en PHP. Dans l'exemple ci-dessus, nous avons affiché un élément de Tableau (Array) . Les tableaux peuvent être très utiles.

`$_SERVER` est juste une variable qui est automatiquement disponible dans votre script. Une liste de toutes les variables qui sont rendues disponibles est fournie dans la section Variables réservées ou vous pouvez aussi en obtenir une liste complète en lisant l'affichage de la fonction `phpinfo` utilisée dans l'exemple de la section précédente.

Vous pouvez ajouter plusieurs commandes PHP dans une balise PHP, et créer de petits blocs de code qui réalisent des opérations plus complexes qu'un simple affichage. Par exemple, si nous voulons vérifier que le navigateur est bien de la famille des Internet Explorer, nous pouvons faire cela :

Exemple utilisant les structures de contrôle et les fonctions

```
<?php
if (strpos($_SERVER['HTTP_USER_AGENT'], 'MSIE') !== FALSE) {
```

```

    echo 'Vous utilisez Internet Explorer<br />';
}
?>

```

Le résultat de ce script, si vous utilisez Internet Explorer, sera :

Vous utilisez Internet Explorer

Ici, nous introduisons plusieurs nouveaux concepts. Nous avons une structure `if`. Si vous êtes familier avec les syntaxes de base du langage C, cela ne vous surprendra pas. Si vous ne connaissez pas assez le langage C ou un autre langage où la syntaxe est similaire à celle ci-dessus, il vaudrait mieux que vous lisiez une introduction à PHP, et assimiliez les premiers chapitres, ou bien lisez le chapitre consacré à [la référence du langage](#). Vous pouvez trouver une liste de livres PHP à <http://www.php.net/books.php>.

Le second concept que nous avons introduit est la fonction `strpos`. `strpos` est une fonction intégrée à PHP, qui recherche la présence d'une chaîne dans une autre. Dans notre cas, nous avons recherché la chaîne "MSIE" dans la chaîne `$_SERVER['HTTP_USER_AGENT']`. Si cette chaîne est trouvée, la fonction retourne sa position dans la chaîne et sinon, `FALSE`. Si elle ne retourne pas `FALSE`, la structure `if` reçoit `TRUE` et le code entre accolades `{}` est exécuté. Sinon, le code n'est pas exécuté. N'hésitez pas à expérimenter d'autres exemples, à l'aide de `if`, `else`, et d'autres fonctions comme `strtoupper` et `strlen`. Chaque page de la documentation contient aussi des exemples. Si vous n'êtes pas sûr de l'utilisation de ces fonctions, vous devez lire la page du manuel "[comment lire une définition de fonction](#)" ainsi que la [section sur les fonctions PHP](#).

Nous pouvons maintenant progresser et vous montrer comment utiliser le mode PHP, au milieu du code HTML :

Passer du mode PHP au mode HTML et vice-versa

```

<?php
if (strpos($_SERVER['HTTP_USER_AGENT'], 'MSIE') !== FALSE) {
?>
<h3>strpos() n'a pas retourné FALSE</h3>
<p>Vous utilisez Internet Explorer</p>
<?php
} else {
?>
<h3>strpos() a retourné FALSE</h3>
<p>Vous n'utilisez pas Internet Explorer</p>
<?php
}
?>

```

Un exemple de résultat obtenu dans ce script est :

```

<h3>strpos() n'a pas retourné FALSE</h3>
<p>Vous utilisez Internet Explorer</p>

```

Au lieu d'utiliser une commande `echo`, pour afficher du texte, vous pouvez utiliser du code HTML pur. Le point important à noter ici est que la logique de programmation est conservée. Seul un des deux blocs HTML sera affiché, suivant le résultat de la fonction `strpos`. En d'autres termes, cela dépend si la chaîne MSIE a été trouvée ou non.

3.2.4 Utiliser un formulaire

L'un des points forts de PHP est sa capacité à gérer les formulaires. Le concept de base qui est important à comprendre est que tous les champs d'un formulaire seront automatiquement disponibles dans le script PHP d'action. Lisez le chapitre du manuel concernant les [variables d'origine externe à PHP](#) pour plus d'informations et d'exemples sur comment utiliser les formulaires. Voici un exemple de formulaire HTML :

Un simple formulaire HTML

```
<form action="action.php" method="post">
  <p>Votre nom : <input type="text" name="nom" /></p>
  <p>Votre age : <input type="text" name="age" /></p>
  <p><input type="submit" value="OK"></p>
</form>
```

Il n'y rien de particulier dans ce formulaire. Il est en HTML pur, sans aucune configuration particulière. Lorsque le visiteur remplit le formulaire, et clique sur le bouton OK, le fichier action.php est appelé. Dans ce fichier, vous pouvez écrire le script suivant :

Afficher des données issues d'un formulaire

```
Bonjour, <?php echo $_POST['nom']; ?>.
Tu as <?php echo $_POST['age']; ?> ans.
```

Voici le résultat que vous pourriez obtenir, selon les valeurs que vous avez saisies :

```
Bonjour Jean.
Tu as 29 ans.
```

Le fonctionnement du script devrait être évident. Il n'y a rien de plus simple. Les variables `$_POST['nom']` et `$_POST['age']` sont automatiquement créés par PHP. Un peu plus tôt dans ce tutoriel, nous avons utilisé la variable `$_SERVER`, une auto-globale. Maintenant, nous avons introduit une autre auto-globale `$_POST` qui contient toutes les données envoyées par la méthode **POST**. Notez que dans notre formulaire, nous avons choisi la méthode **POST**. Si vous avons utilisé la méthode **GET** alors notre formulaire aurait placé ses informations dans la variable `$_GET`, une autre auto-globale. Vous pouvez aussi utiliser la variable `$_REQUEST`, si vous ne souhaitez pas vous embarrasser de la méthode utilisée. Elle contient un mélange des données de GET, POST, COOKIE et FILE. Voyez aussi la fonction [import_request_variables](#).

Vous pouvez également utiliser champs XForms dans PHP, même si vous vous sentez bien dans l'utilisation des formulaires HTML. Bien que le travail avec XForms ne soit pas fait pour les débutants, vous pourriez être intéressé par cette technologie. Nous avons également une [courte introduction sur le traitement des données reçues par XForms](#) dans notre section sur les fonctionnalités.

3.2.5 Utiliser des codes anciens avec les nouvelles versions de PHP

Maintenant que PHP est devenu un langage de script populaire, il existe de nombreuses ressources qui vous proposent des portions de code que vous pouvez réutiliser dans vos codes. Pour la plupart, les développeurs de PHP ont tâché d'assurer la compatibilité ascendante, ce qui fait que de nombreux scripts écrits pour les anciennes versions sont aussi valables pour les nouvelles versions de PHP, idéalement sans modifications. En pratique, certaines modifications doivent être apportées.

Les deux modifications récentes les plus importantes qui affectent les anciens codes sont :

- Les anciennes variables `$HTTP_*_VARS` (qui devaient être indiquées comme globales pour être utilisées dans une fonction ou une méthode) sont obsolètes. Les nouveaux tableaux auto-globaux ont été introduits en [PHP 4.1.0](#) . Ce sont les variables suivantes : `$_GET` , `$_POST` , `$_FILES` , `$_COOKIE` , `$_SERVER` , `$_ENV` , `$_REQUEST` et `$_SESSION` . Les vieux tableaux `$HTTP_*_VARS` , tels que `$HTTP_POST_VARS` existent toujours depuis PHP 3. Depuis PHP 5.0.0, les tableaux prédéfinis PHP peuvent être désactivés avec l'option de configuration register_long_arrays .
- Les variables externes ne sont plus enregistrées dans l'environnement par défaut. En d'autres termes, depuis PHP [4.2.0](#) , la directive PHP register_globals vaut **off** par défaut dans le `php.ini` . La méthode recommandée pour accéder à ces valeurs, est via les tableaux auto-globaux mentionnés ci-dessus. Les anciens scripts, livres et tutoriaux continuent de considérer que cette directive devrait être à on . Lorsque cette directive est à on , vous pouvez utiliser la variable `$id` , si l'URL `http://www.example.com/foo.php?id=42` a été appelée. Quelle que soit la valeur de la directive, `$_GET['id']` est toujours disponible.

Pour plus de détails sur ces modifications, reportez-vous à la section sur les variables pré-définies .

3.2.6 Et après?

Avec ce que vous savez, vous êtes maintenant capable de comprendre l'essentiel de la documentation PHP, et les différents scripts d'exemples disponibles dans les archives. Vous pouvez aussi trouver d'autres exemples dans la section liens ("links", en anglais) du site <http://www.php.net/links.php> .

Différentes présentations des capacités de PHP sont disponibles sur le site des conférences PHP : <http://conf.php.net/> et <http://talks.php.net/> .

4 Installation et configuration

4.1 Considérations générales sur l'installation

Avant d'installer PHP, vous devez savoir ce que vous voulez faire de PHP. Il y a trois cas d'utilisation que vous a décrit la section [Que peut faire?](#) :

- Scripts web
- Scripts de ligne de commande
- Application à interface graphique

Pour la première tâche, qui est de loin la plus répandue, vous avez besoin de trois choses : PHP lui-même, un serveur web et un navigateur. Vous avez probablement un navigateur, et en fonction de votre système d'exploitation, vous pouvez aussi disposer d'un serveur Web (i.e. Apache sous Linux ou IIS sous Windows). Vous pouvez aussi louer un espace à une société. De cette façon, vous n'aurez pas à mettre en place PHP, mais uniquement à écrire vos scripts, les charger sur le serveur et voir le résultat sur votre navigateur.

Si vous installez PHP et le serveur par vous-même, vous avez deux choix. Soit sous la forme d'un module du serveur web (via une interface directe appelée SAPI). Les serveurs qui supportent cette solution comptent notamment Apache, Microsoft Internet Information Server, Netscape et iPlanet. D'autres serveurs ont aussi le support ISAPI, l'interface de module Microsoft (OmniHTTPd par exemple). Si PHP ne supporte pas l'interface de module de votre serveur web, vous pouvez toujours l'utiliser comme processeur CGI. Cela signifie que vous devez configurer votre serveur pour qu'il utilise l'exécutable PHP (php.exe sous Windows), pour qu'il traite les fichiers PHP sur le serveur.

Si vous souhaitez aussi utiliser PHP en ligne de commande (écrire des scripts de génération d'image hors ligne, par exemple, ou bien traiter des textes en fonctions d'information que vous leur passeriez), vous aurez besoin d'un exécutable PHP. Pour plus de détails, lisez la section [écrire des applications PHP en ligne de commande](#) . Dans ce cas, vous n'aurez pas besoin de serveur web, ni de navigateur.

Avec PHP, vous pouvez aussi écrire des interfaces graphiques clientes, en utilisant l'extension PHP-GTK. C'est une approche complètement différente de l'écriture des pages web, car vous ne générerez pas de fichiers HTML, mais vous aurez à gérer des fenêtres et des objets. Pour plus de détails sur PHP-GTK, voyez le [site officiel](#) . PHP-GTK n'est pas inclus dans la distribution officielle de PHP.

A partir de maintenant, cette section décrit l'installation de PHP avec un serveur web sous Unix et Windows, sous forme de module ou d'exécutables CGI.

Les codes source et les exécutables compilés de certains OS (y compris Windows), sont disponibles à <http://www.php.net/downloads.php> . Nous recommandons l'utilisation du [miroir](#) le plus proche pour accélérer les chargements.

4.2 Installation sous UNIX

Cette section va vous guider lors du processus d'installation et de configuration de PHP sous Unix. Commencez par étudier les sections spécifiques à votre plate-forme ou à votre serveur web avant de passer à l'installation.

Comme ce que nous avons écrit dans le manuel dans la section [Considérations générales sur l'installation](#) , nous traiterons de l'installation de PHP sur des serveurs web dans cette section, bien que nous couvrirons également la configuration de PHP pour l'utilisation en lignes de commande.

Il y a plusieurs manières d'installer PHP sur des plates-formes Unix, soit avec un processus de compilation / configuration, soit avec des paquets pré-compilés. Cette documentation est particulièrement focalisée sur le processus de compilation / configuration. Beaucoup de systèmes basés sur Unix ont plusieurs sortes de paquets d'installation pour leur système. Ils permettent de vous assister dans une configuration standard mais si vous avez besoin d'avoir des fonctionnalités différentes (comme un serveur sécurisé ou un driver différent de bases de données), vous aurez besoin de construire PHP et/ou votre serveur web. Si vous n'êtes pas familiarisé avec la construction et la compilation de vos propres logiciels, il sera plus simple de vérifier si quelque part, personne n'a déjà construit une version de paquet de PHP avec les fonctionnalités dont vous avez besoin.

Pré-requis :

- Connaissance de base d'UNIX (savoir faire un "make" et compiler en C, si besoin).
- Un compilateur ANSI C (pour les codes sources)
- flex: Version 2.5.4 (pour compiler)
- bison: Version 1.28 (préférée), 1.35, or 1.75 (pour compiler)
- Un serveur web
- Tous les composants nécessaires aux extensions (bibliothèque GD , PDF , etc)

La configuration initiale de PHP et le processus de configuration sont contrôlés par l'utilisation du fichier configure et de ces options en lignes de commande. Vous pouvez récupérer une liste de toutes les options disponibles accompagnées d'une courte description en exécutant la commande `./configure --help` . Notre manuel documente les différentes options séparément. Vous pouvez trouver les [options internes en annexe](#) , bien que les différentes options spécifiques à chaque extension sont décrites sur les pages de référence.

Lorsque PHP est configuré, vous êtes prêt à construire le module et/ou l'exécutable. La commande make devrait s'occuper de cela. Si elle échoue et que vous ne savez pas pourquoi, lisez la [section Problèmes](#) .

4.2.1 Apache 1.3.x sur les systèmes Unix

Cette section contient des notes spécifiques pour l'installation de PHP avec Apache sur les systèmes Unix. Des notes spécifiques pour Apache 2 sont aussi disponibles [sur une page séparée](#) .

Vous pouvez sélectionner des options à ajouter au fichier configure à la ligne 10 depuis la [liste complète des options de configuration](#) . Les numéros de versions ont été omis ici afin de s'assurer que les instructions ne soient pas incorrectes. Vous devrez donc remplacer les 'xxx' par les versions correctes de vos fichiers.

Instructions d'installation de PHP (en module Apache)

1. `gunzip apache_XXX.tar.gz`
2. `tar -xvf apache_XXX.tar`
3. `gunzip php-XXX.tar.gz`
4. `tar -xvf php-XXX.tar`
5. `cd apache_XXX`
6. `./configure --prefix=/www --enable-module=so`

7. make
8. make install
9. cd ../php-xxx

10. Maintenant, configurez votre PHP. C'est l'endroit où vous configurez PHP avec diverses options, comme les extensions qui seront activées. Lancez `./configure --help` pour une liste des options disponibles. Dans notre exemple, nous ferons un `./configure` assez simple avec uniquement le support Apache et MySQL. Votre chemin vers apxs peut être différent de notre exemple.

```
./configure --with-mysql --with-apxs=/www/bin/apxs
```

11. make
12. make install

Si vous décidez de changer vos options de configuration après l'installation, vous aurez juste besoin de répéter les trois dernières étapes. Vous aurez aussi besoin de redémarrer apache pour que le nouveau module soit chargé. Une recompilation de Apache n'est pas nécessaire.

Notez que, à moins de l'avoir explicitement désactivé, 'make install' installera aussi PEAR, et des outils PHP tels que phpize, installera le CLI PHP, etc.

13. Configurez votre fichier php.ini :

```
cp php.ini-dist /usr/local/lib/php.ini
```

Vous pouvez éditer votre fichier .ini pour régler certaines options PHP. Si vous souhaitez votre php.ini à un autre endroit, utilisez `--with-config-file-path=/votre/chemin` lors de l'étape 10.

Si vous utilisez plutôt `php.ini-recommended`, assurez-vous de lire l'ensemble des changements qui y sont contenus, car ils modifient le fonctionnement de PHP.

14. Editez votre `httpd.conf` afin de charger le module PHP. Le chemin dans la partie droite de la directive `LoadModule` doit pointer vers l'endroit où se trouve le module PHP sur votre système. Le `make install` lancé plus haut l'y aura certainement déjà déposé pour vous, mais assurez le vous.

Pour PHP 4 :

```
LoadModule php4_module libexec/libphp4.so
```

Pour PHP 5 :

```
LoadModule php5_module libexec/libphp5.so
```

15. Et dans la section `AddModule` de `httpd.conf`, quelque part en dessous de `ClearModuleList`, ajoutez ceci :

Pour PHP 4 :

```
AddModule mod_php4.c
```

Pour PHP 5 :

```
AddModule mod_php5.c
```

16. Dites à Apache de faire parser certaines extensions par PHP. Par exemple, faites parser l'extension .php par PHP. Vous pouvez ajouter n'importe quelle(s) extension(s) à parser juste en les ajoutant à la suite, séparées par un espace. Nous ajouterons .phtml dans notre exemple.

```
AddType application/x-httpd-php .php .phtml
```

Il est assez fréquent de configurer l'extension .phps comme code source PHP colorisé, ce qui peut être fait ainsi :

```
AddType application/x-httpd-php-source .phps
```

17. Utilisez votre méthode habituelle pour démarrer le serveur Apache. (vous devez l'éteindre et le redémarrer, pas seulement lui envoyer un signal HUP ou USR1.)

Alternativement, pour installer PHP en tant qu'objet statique :

Instructions d'installation (Installation en tant que module statique d'Apache) de PHP

```
1. gunzip -c apache_1.3.x.tar.gz | tar xf -
```

```
2. cd apache_1.3.x
```

```
3. ./configure
```

```
4. cd ..
```

```
5. gunzip -c php-4.x.y.tar.gz | tar xf -
```

```
6. cd php-4.x.y
```

```
7. ./configure --with-mysql --with-apache=../apache_1.3.x
```

```
8. make
```

```
9. make install
```

```
10. cd ../apache_1.3.x
```

```
11. ./configure --prefix=/www --activate-module=src/modules/php4/libphp4.a
```

(La ligne ci-dessus est correct ! Oui, nous savons que libphp4.a n'existe pas à ce moment. On ne le suppose pas non plus. Il sera créé.)

```
12. make
```

(vous devriez avoir maintenant un binaire httpd que vous pouvez copier dans votre dossier de binaire Apache ; si c'est votre première installation, vous devez exécuter la commande "make install")

```
13. cd ../php-4.x.y
```

```
14. cp php.ini-dist /usr/local/lib/php.ini
```

15. Vous pouvez éditer le fichier /usr/local/lib/php.ini pour définir les options de PHP.

Editez votre fichier httpd.conf ou srm.conf et ajoutez :

```
AddType application/x-httpd-php .php
```

Note

Remplacez php-4 par php-5 et php4 par php5 en PHP 5.

Suivant votre installation d'Apache et votre variante d'Unix, il existe de nombreuses façons d'arrêter

et redémarrer Apache. Voici une liste des commandes typiques, pour différentes installations. Remplacez /path/to/ par le chemin d'accès à vos applications sur votre système.

Exemples de commandes pour le redémarrage d'apache

```
1. Nombreuses variantes Linux SysV :
/path/to/httpd restart

2. Avec les scripts apachectl :
/path/to/apachectl stop
/path/to/apachectl start

3. httpdctl et httpsdctl (utilisant OpenSSL), similaires à apachectl:
/path/to/httpsdctl stop
/path/to/httpsdctl start

4. En utilisant mod_ssl, ou un autre serveur SSL, vous pouvez vouloir l'arrêter
et le démarrer manuellement :
/path/to/apachectl stop
/path/to/apachectl startssl
```

L'emplacement des exécutables apachectl et http(s)dctl peut varier. Si votre système est pourvu des commandes locate , whereis ou which , elles peuvent vous aider à retrouver vos programmes.

Différents exemples de compilation PHP pour Apache suivent :

```
./configure --with-apxs --with-pgsql
```

Cette commande va créer une bibliothèque partagée libphp4.so qui sera chargée par Apache avec une ligne LoadModule dans le fichier httpd.conf . Le support PostgreSQL est aussi inclut dans libphp4.so .

Compilation de PHP avec le support PostGreSQL

```
./configure --with-apxs --with-pgsql=shared
```

Cette commande va créer une bibliothèque partagée libphp4.so pour Apache, mais va aussi créer la bibliothèque partagée pgsql.so qui sera chargée dans PHP avec une directive du fichier php.ini file ou en la chargeant explicitement dans le script avec la fonction `dll` .

Compilation en bibliothèques statiques

```
./configure --with-apache=/path/to/apache_source --with-pgsql
```

Cette commande va créer une autre bibliothèque partagée libmodphp4.a , un fichier mod_php4.c et quelques fichiers associés dans le dossier src/modules/php4 du dossier source Apache. Puis, vous devez compiler Apache avec `--activate-module=src/modules/php4/libphp4.a` et le système de compilation d'Apache va créer un fichier libphp4.a et le lier statiquement avec httpd (remplacez php4 par php5 en PHP 5). Le support PostgreSQL est alors inclus directement dans l'exécutable httpd , ce qui fait que le résultat final est un fichier unique httpd , qui inclut Apache et PHP.

Compilation en bibliothèques partagées

```
./configure --with-apache=/path/to/apache_source --with-pgsql=shared
```

Comme précédemment, mais au lieu d'inclure le support PostgreSQL directement dans l'exécutable httpd final, vous allez obtenir une bibliothèque partagée pgsql.so que vous pouvez charger dans PHP soit grâce au fichier de configuration php.ini ou dynamiquement avec `dll` .

Lorsque vous faites votre choix entre les différents modes de compilation de PHP, vous devez prendre en compte leurs avantages et inconvénients respectifs. Les objets partagés permettent de compiler PHP et Apache de manière séparée, et vous n'aurez pas à compiler l'ensemble pour faire évoluer PHP. La compilation statique permet de charger et d'exécuter plus rapidement PHP. Pour plus d'informations, voyez la page web sur le [support des DSO](#) .

Note

Le httpd.conf par défaut de Apache est fourni avec une section qui ressemble à ceci :

httpd.conf fournit par Apache

```
User nobody
Group "#-1"
```

A moins que vous ne changiez cette valeur par "Group nogroup" ou quelque chose comme ça ("Group daemon" est aussi classique), PHP ne sera pas capable d'ouvrir des fichiers.

Note

Assurez-vous que vous spécifiez la version installée de apxs avec l'option --with-apxs=/path/to/apxs . Vous ne devez PAS utiliser la version d'apxs qui est dans les sources d'Apache, mais celle qui est réellement installée sur votre système.

4.2.2 Apache 2.0 sur les systèmes Unix

Cette section contient les notes et conseils d'installation de PHP avec le serveur Apache 2.0 sur les systèmes Unix.

Attention

Nous ne recommandons pas l'utilisation de PHP dans un environnement threadé MPM, avec Apache 2. Utilisez le mode prefork MPM à la place, ou utilisez Apache 1. Pour savoir pourquoi, lisez l'entrée de la FAQ correspondante à l' [utilisation d'Apache 2 dans un environnement threadé MPM](#) .

Il est très recommandé de lire la [documentation Apache](#) pour avoir une meilleure connaissance du serveur web Apache 2.0.

Téléchargez la version la plus récente de [Apache 2.0](#) et une version appropriée de PHP, tel que décrit ci-dessus. Ce guide rapide couvre les manipulations de base, nécessaires à l'installation de Apache 2.0 et PHP. Pour plus d'informations, lisez la [documentation Apache](#) . Les numéros de version sont omis ici, pour s'assurer que les instructions ne soient pas incorrectes. Il faudra donc remplacer les séquences 'NN' avec les valeurs correctes que vous utilisez.

Instruction d'installation (Module partagé Apache 2)

1. gzip -d httpd-2_0_NN.tar.gz
2. tar xvf httpd-2_0_NN.tar
3. gunzip php-NN.tar.gz
4. tar -xvf php-NN.tar
5. cd httpd-2_0_NN
6. ./configure --enable-so
7. make
8. make install

Maintenant, vous avez un dossier Apache 2.0.NN installé dans /usr/local/apache2, configure avec le support des modules dynamiques, et le prefork standard MPM. Pour tester l'installation, utilisez votre procédure standard de démarrage d'Apache :
 /usr/local/apache2/bin/apachectl start
 et pour stopper le serveur, utilisez
 /usr/local/apache2/bin/apachectl stop.

9. cd ../php4-NN

10. Maintenant, configurez votre PHP. C'est l'endroit où vous configurez PHP avec diverses options, comme les extensions qui seront activées. Lancez `./configure --help` pour une liste des options disponibles. Dans notre exemple, nous ferons une configuration assez simple avec uniquement le support Apache 2 et MySQL. Votre chemin vers apxs peut être différent, en fait, le binaire devrait toujours s'appeler `apxs2` sur votre système.

```
./configure --with-apxs2=/usr/local/apache2/bin/apxs --with-mysql
```

11. `make`

12. `make install`

Si vous décidez de changer vos options de configuration après l'installation, vous aurez juste besoin de répéter les trois dernières étapes. Vous aurez aussi besoin de redémarrer apache pour que le nouveau module soit chargé. Une recompilation de Apache n'est pas nécessaire.

If you decide to change your configure options after installation, you only need to repeat the last three steps. You only need to restart apache for the new module to take effect. A recompile of Apache is not needed.

Notez que, à moins de l'avoir explicitement désactivé, 'make install' installera aussi PEAR, et des outils PHP tels que `phpize`, installera le CLI PHP, etc.

13. Configurez votre fichier `php.ini` :

```
cp php.ini-dist /usr/local/lib/php.ini
```

Vous pouvez éditer votre fichier `.ini` pour régler certaines options PHP. Si vous souhaitez votre `php.ini` à un autre endroit, utilisez `--with-config-file-path=/votre/chemin` lors de l'étape 10.

Si vous utilisez plutôt `php.ini-recommended`, assurez-vous de lire l'ensemble des changements qui y sont contenus, car ils modifient le fonctionnement de PHP.

14. Éditez votre `httpd.conf` afin de charger le module PHP. Le chemin dans la partie droite de la directive `LoadModule` doit pointer vers l'endroit où se trouve le module PHP sur votre système. Le `make install` lancé plus haut l'y aura certainement déjà déposé pour vous, mais assurez-le vous.

Pour PHP 4 :

```
LoadModule php4_module modules/libphp4.so
```

Pour PHP 5:

```
LoadModule php5_module modules/libphp5.so
```

15. Dites à Apache de faire parser certaines extensions par PHP. Par exemple, faites parser l'extension `.php` par PHP. Vous pouvez ajouter n'importe quelle(s) extension(s) à parser juste en les ajoutant à la suite, séparées par un espace. Nous ajouterons `.phtml` dans notre exemple.

```
AddType application/x-httpd-php .php .phtml
```

Il est assez fréquent de configurer l'extension .phps comme code source PHP colorisé, ce qui peut être fait ainsi :

```
AddType application/x-httpd-php-source .phps
```

16. Utilisez votre méthode habituelle pour démarrer le serveur Apache, e.g.:

```
/usr/local/apache2/bin/apachectl start
```

Suivez les étapes ci-dessus, et vous disposerez d'un serveur Apache 2.0 avec le support de PHP module comme module SAPI . Bien sur, il y a bien d'autres options de configuration disponibles pour les deux logiciels, Apache et PHP. Pour plus de détails, utilisez la commande `./configure --help` dans le dossier de sources approprié. Si vous souhaitez compiler une version multi-threadé de Apache 2.0 vous devrez remplacer le module standard MPM prefork avec worker ou perchild . Pour ce faire, ajoutez à la ligne de configuration de l'étape 6, l'option `--with-mpm=worker` ou `--with-mpm=perchild` . Soyez conscient des conséquences, et comprenez bien ce que vous faites. Pour plus de détails sur ce sujet, lisez la documentation Apache sur [le module MPM](#) .

Note

Si vous voulez utiliser la négociation sur le contenu, lisez la section [FAQ Apache MultiViews](#) .

Note

Pour compiler une version multi-threadée d'Apache, votre système doit supporter les threads. Cela implique aussi de compiler PHP avec le module expérimental de Zend Thread Safety (ZTS). Par conséquent, toutes les extensions ne seront pas disponibles. La configuration recommandée actuellement est celle avec le module standard MPM prefork .

4.2.3 Caudium

PHP 4 peut être compilé comme module Pike pour [le serveur web Caudium](#) . Notez que ce mode n'est pas supporté en PHP 3. Suivez simplement les instructions suivantes pour installer PHP 4 sur un serveur Caudium.

Instructions d'installation Caudium

1. Assurez-vous que vous avez un serveur Caudium installé avant de tenter l'installation PHP 4. Pour que PHP 4 fonctionne correctement, vous devez installer Pike 7.0.268 ou plus récent. Pour cet exemple, nous supposons que vous avez installé Caudium dans le dossier `/opt/caudium/server/`.
2. Renommez le dossier en `php-x.y.z` (où `x.y.z` est le numéro de version).
3. `./configure --with-caudium=/opt/caudium/server`
4. `make`
5. `make install`
6. Redémarrez Caudium s'il était en fonctionnement.
7. Connectez-vous à l'interface de configuration graphique et allez dans le serveur virtuel auquel vous voulez ajouter le support PHP 4.
8. Cliquez sur "Add Module" et recherchez puis ajoutez le module "PHP 4 Script Support".
9. Si la documentation dit que 'PHP 4 interpreter isn't available', assurez-vous que vous avez bien redémarré le serveur. Si vous l'avez fait, vérifiez le fichier `/opt/caudium/logs/debug/default.1` : il contient peut-être des erreurs liées à PHP4.so. De même, assurez-vous

que caudium/server/lib/[pike-version]/PHP4.so est présent.

10. Configurez le module "PHP Script Support" si nécessaire.

Vous pouvez bien sûr compiler votre module Caudium avec les diverses extensions disponibles. Voyez la page de référence pour la liste des options de configuration.

Note

Lorsque vous ajoutez le support MySQL à PHP 4, vous devez vous assurer que le client MySQL normal est utilisé. Sinon, il peut y avoir des conflits avec Pike, qui dispose déjà du support MySQL. Vous pouvez le faire en spécifiant le dossier d'installation de MySQL grâce à l'option `--with-mysql`.

4.2.4 Installation avec les serveurs fhttpd

Pour compiler PHP comme un module fhttpd, répondre "yes" à la question "Build as an fhttpd module ?" (cela correspond à l'option de configuration `--with-fhttpd = DIR` et spécifier la racine de la distribution fhttpd. Le répertoire par défaut est `/usr/local/src/fhttpd`. Si vous utilisez fhttpd, compiler PHP en module vous permettra d'obtenir des performances supérieures, plus de contrôle et la possibilité d'exécution à distance.

Note

Le support de fhttpd n'est plus disponible depuis PHP 4.3.0.

4.2.5 Installation sous Netscape et iPlanet Enterprise Serveur sur un système Sun Solaris

Cette section contient les notes et détails spécifiques à l'installation de PHP sur les serveurs web Sun Java System Web Server, Sun ONE Web Server, iPlanet et Netscape server sur les systèmes Sun Solaris.

Depuis PHP 4.3.3, vous pouvez utiliser les scripts PHP avec le module NSAPI pour gérer des listes de dossiers et des pages d'erreurs personnalisées. Des fonctions supplémentaires sont disponibles pour assurer la compatibilité avec Apache. Pour du support sur les serveurs courants, voyez la note sur les sous-requêtes.

Vous pouvez trouver plus d'informations sur la configuration de PHP avec Netscape Enterprise Server : <http://benoit.noss.free.fr/php/install-php4.html>

Pour construire PHP avec les serveurs web Sun JSWS/Sun ONE WS/iPlanet/Netscape, entrez le répertoire valide d'installation pour l'option `--with-nsapi=[DIR]`. Le répertoire par défaut est habituellement `/opt/netscape/suitespot/`. Lisez également le fichier `/php-xxx-version/sapi/nsapi/nsapi-readme.txt`.

1. Installez les packages suivants depuis le serveur <http://www.sunfreeware.com/> ou un miroir ad hoc :

- ◆ autoconf-2.13
- ◆ automake-1.4
- ◆ bison-1_25-sol26-sparc-local
- ◆ flex-2_5_4a-sol26-sparc-local
- ◆ gcc-2_95_2-sol26-sparc-local
- ◆ gzip-1.2.4-sol26-sparc-local
- ◆ m4-1_4-sol26-sparc-local
- ◆ make-3_76_1-sol26-sparc-local
- ◆ mysql-3.23.24-beta (si vous voulez le support MySQL)

- ◆ perl-5_005_03-sol26-sparc-local
 - ◆ tar-1.13 (GNU tar)
2. Assurez-vous que le path inclut bien les dossiers nécessaires :
PATH=./usr/local/bin:/usr/sbin:/usr/bin:/usr/ccs/bin et rendez le accessible à votre système avec export PATH .
 3. gunzip php-x.x.x.tar.gz (si vous avez une distribution .gz , ou bien allez en 4).
 4. tar xvf php-x.x.x.tar
 5. Passez dans votre dossier PHP : cd ../php-x.x.x
 6. Pour les étapes suivantes, assurez-vous que /opt/netcape/suitespot/ correspond bien à votre installation du serveur netcape. Sinon, indiquez le chemin correct :

```
./configure --with-mysql=/usr/local/mysql \  
--with-nsapi=/opt/netcape/suitespot/ \  
--enable-libgcc
```

7. Faites un make puis un make install .

Après avoir fait l'installation de base et lu les fichiers readme.txt, vous pouvez avoir besoin de faire des configurations supplémentaires.

Instructions de configuration pour Sun/iPlanet/Netscape

Tout d'abord, vous aurez besoin d'ajouter des chemins dans la variable LD_LIBRARY_PATH pour que Netscape trouve son bonheur. Il est préférable de le faire dans le script de démarrage du serveur Netscape. Les utilisateurs Windows peuvent ignorer cette étape. Le script de démarrage est souvent situé dans : /path/to/server/https-servername/start . Vous aurez peut être à éditer le fichier de configuration situé dans /path/to/server/https-servername/config/ .

1. Ajoutez les lignes suivantes dans mime.types en tant qu'administrateur :
type=magnus-internal/x-httpd-php exts=php
2. Editez le fichier magnus.conf (pour les serveurs >= 6) ou le fichier obj.conf (pour les serveurs < 6) et ajoutez les lignes suivantes dans obj.conf . shlib peut varier en fonction de votre OS. Pour Unix, c'est quelquechose comme /opt/netcape/suitespot/bin/libphp4.so . Il est conseillé de placer les lignes suivantes après les lignes de mime types init .

```
Init fn="load-modules" funcs="php4_init,php4_execute,php4_auth_trans" shlib="/opt/netcape  
Init fn="php4_init" LateInit="yes" errorString="Failed to initialize PHP!" [php_ini="/path
```

(PHP >= 4.3.3) Le paramètre php_ini est optionnel mais avec lui, vous pouvez placer votre php.ini dans le dossier de configuration de votre serveur web.

3. Configurez l'objet par défaut dans le fichier obj.conf (pour les classes de serveur virtuel [version 6.0+] dans leur fichier vserver.obj.conf) :

```
<Object name="default">  
.  
.  
.  
.#NOTE this next line should happen after all 'ObjectType' and before all 'AddLog' lines  
Service fn="php4_execute" type="magnus-internal/x-httpd-php" [inikey=value inikey=value ..  
.  
.  
</Object>
```

(PHP >= 4.3.3) Comme paramètres additionnels, vous pouvez ajouter quelques valeurs spéciales dans le php.ini , par exemple, vous pouvez définir un spécifique docroot="/path/to/docroot" où le contexte php4_execute est appelé. Pour les init-keys booléens, utilisez les valeurs 0/1, et non pas "On","Off",... (cela ne fonctionnera pas correctement), e.g. zlib.output_compression=1 au lieu de zlib.output_compression="On"

4. Cela est nécessaire uniquement si vous voulez configurer un répertoire pour accueillir des scripts PHP tout comme un répertoire cgi-bin :

```
<Object name="x-httpd-php">
ObjectType fn="force-type" type="magnus-internal/x-httpd-php"
Service fn=php4_execute [inikey=value inikey=value ...]
</Object>
```

Après cela, vous pouvez configurer un répertoire dans le serveur d'administration et y assigner le style x-httpd-php . Tous les fichiers dans ce répertoire seront exécutés comme étant du PHP. C'est pratique pour cacher l'usage de PHP en renommant les fichiers en .html

5. Configuration de l'identification : les identifications PHP ne peuvent pas être utilisées avec aucune autre identification. TOUTES LES IDENTIFICATIONS SONT PASSEES A VOS SCRIPTS PHP. Pour configurer l'identification PHP pour tout le serveur web, ajoutez la ligne suivante à votre objet par défaut :

```
<Object name="default">
AuthTrans fn=php4_auth_trans
.
.
.
</Object>
```

6. Pour utiliser l'identification PHP dans un seul répertoire, ajoutez ce qui suit :

```
<Object ppath="d:\path\to\authenticated\dir\*">
AuthTrans fn=php4_auth_trans
</Object>
```

Note

La taille de la pile que PHP utilise dépend de la configuration du serveur web. Si vous rencontrez des crashes avec les grands scripts PHP, il est recommandé d'augmenter la taille de la pile avec la console d'administration : dans la section "MAGNUS EDITOR".

4.2.5.1 Environnement CGI et modifications recommandées du php.ini

Il est important de garder en tête que iPlanet/SunONE/Netscape est un serveur web multi-threadé. Comme toutes les requêtes se situent dans le même contexte (c'est le contexte sur serveur web), et que ce contexte est unique. Si vous voulez accéder à des variables comme PATH_INFO , HTTP_HOST etc. il n'est pas recommandé d'y accéder à la manière de PHP 3.x, avec la fonction getenv ou une autre méthode (register_globals, \$_ENV). De cette manière, vous n'aurez que des valeurs d'environnement du serveur, et non pas des valeurs correctes pour le CGI.

Note

Pourquoi est-ce que les variables CGI sont invalides ?

C'est lié au fait que le processus du serveur web est lancé par l'administrateur du serveur, qui utilise le script de lancement au démarrage. En fait, il aurait fallu que vous lanciez vous-même le processus. C'est pour cela que l'environnement du serveur web contient des variables d'environnement CGI. Vous pouvez vérifier cela en lançant le serveur web depuis un autre endroit que l'administrateur du serveur : utilisez la ligne de commande Unix en tant que root : vous verrez alors qu'il n'y a pas de variables d'environnement.

Changez simplement vos scripts pour lire les variables CGI, en utilisant le tableau superglobal \$_SERVER . Si vous avez d'autres scripts qui utilisent encore \$HTTP_HOST et compagnie, il est recommandé d'activer l'option register_globals dans le php.ini et de changer l'ordre des variables. IMPORTANT : supprimez le "E" dans cette option, car vous n'en avez pas besoin pour cet

environnement.

```
variables_order = "GPCS"
register_globals = On
```

4.2.5.2 Utilisation particulière pour les pages d'erreurs ou les listages spécifiques de dossier (PHP >= 4.3.3)

Vous pouvez utiliser PHP pour générer des pages d'erreurs de type "404 Not Found" ou apparantée. Ajoutez la ligne suivante dans le fichier obj.conf pour chaque page d'erreur que vous souhaitez remplacer :

```
Error fn="php4_execute" code=XXX script="/chemin/vers/script.php" [inikey=value inikey=value...]
```

où XXX est le code d'erreur HTTP. Effacez toute autre directive Error qui pourrait interférer avec la votre. Si vous voulez utiliser une page pour toutes les erreurs qui existent, laissez le paramètre code vide. Votre script peut obtenir le code de statut HTTP dans la variable `$_SERVER['ERROR_TYPE']`.

Une autre possibilité est de générer une liste de dossier personnalisée. Créez simplement un script PHP qui affiche le contenu du dossier, et remplacez la ligne Service par défaut par `type="magnus-internal/directory"` dans obj.conf avec ceci :

```
Service fn="php4_execute" type="magnus-internal/directory" script="/chemin/vers/script.php" [inik
```

Pour ces deux points, l'URI originale et l'URI traduite sont dans les variables `$_SERVER['PATH_INFO']` et `$_SERVER['PATH_TRANSLATED']`.

4.2.5.3 Note au sujet de `nsapi_virtual` et des requêtes (PHP >= 4.3.3)

Le module NSAPI supporte désormais la fonction `nsapi_virtual` (alias : `virtual`), pour réaliser des sous requêtes au serveur web, et inclure le résultat dans une page. Le problème est que cette fonction utilise une fonctionnalité non documentée de la bibliothèque NSAPI. Sous Unix, ce n'est pas un problème, car le module va automatiquement rechercher les fonctions nécessaires, et les utiliser si elles sont disponibles. Sinon, `nsapi_virtual` sera désactivée.

Note

Soyez prévenu : le support de <code>nsapi_virtual</code> est EXPERIMENTAL !!!

4.2.6 CGI et configurations en ligne de commande

Par défaut, PHP est compilé en version CGI. Cela crée un interpréteur de commande qui peut être utilisé soit pour le traitement CGI, soit pour les scripts non relatifs au web. Si PHP peut être incorporé au serveur web que vous utilisez en tant que module, de manière générale c'est cette solution que vous devriez adopter pour des raisons de performances. Cependant, la version CGI permet aux utilisateurs sous Apache de lancer des scripts PHP sous leurs UID respectives.

Attention

En utilisant le mode CGI, votre serveur est ouvert à de possibles attaques sérieuses. Lisez attentivement notre section sur la sécurité en mode CGI pour apprendre comment vous défendre contre ces attaques.

Avec PHP 4.3.0, d'importants ajouts ont été faits à PHP. une nouvelle SAPI, appelée CLI, existe aussi et porte le même nom que la version CGI. Ce qui est installé en tant que {PREFIX}/bin/php dépend de votre ligne de configuration. Tout ceci est décrit en détails dans la partie du manuel intitulée Utiliser PHP en ligne de commande . Veuillez-vous y référer pour de plus amples informations.

4.2.6.1 Tests

Si vous avez compilé PHP comme programme CGI, vous pouvez tester votre produit en tapant : `make test` . C'est toujours une bonne chose de tester le résultat d'une compilation. Cela vous permet de repérer des problèmes entre PHP et votre plate-forme, plutôt que d'attendre qu'ils surviennent.

4.2.6.2 Performances

Si vous avez compilé PHP 3 comme programme CGI, vous pouvez évaluer ses performances avec la commande `make bench` . Notez que si le safe mode est activé (par défaut), vous risquez de voir l'évaluation s'arrêter une fois les 30 secondes réglementaires écoulées. En effet, la fonction set_time_limit ne peut pas être utilisée si le safe mode fonctionne. Utilisez l'option max_execution_time pour contrôler le temps d'exécution de vos scripts. `make bench` ignore le fichier de configuration file .

Note

<code>make bench</code> n'est disponible qu'en PHP 3.

4.2.6.3 Utiliser les variables pré-définies

Certaines variables d'environnement fournies par les serveurs web ne sont pas disponibles dans les spécifications CGI/1.1 actuelles. Seules les variables suivantes sont définies, et les autres doivent être considérées comme spécifiques aux serveurs web : `AUTH_TYPE` , `CONTENT_LENGTH` , `CONTENT_TYPE` , `GATEWAY_INTERFACE` , `PATH_INFO` , `PATH_TRANSLATED` , `QUERY_STRING` , `REMOTE_ADDR` , `REMOTE_HOST` , `REMOTE_IDENT` , `REMOTE_USER` , `REQUEST_METHOD` , `SCRIPT_NAME` , `SERVER_NAME` , `SERVER_PORT` , `SERVER_PROTOCOL` et `SERVER_SOFTWARE` .

4.2.7 Installation sous HP-UX

Cette section contient les notes et conseils d'installation de PHP sur les distributions HP-UX. (Fournit par paul_mckay at clearwater-it dot co dot uk).

Note

Ces conseils sont destinés à PHP 4.0.4 et Apache v1.3.9.
--

1. Vous aurez besoin de gzip. Téléchargez la distribution compilée à <http://hpux.connect.org.uk/ftp/hpux/Gnu/gzip-1.2.4a/gzip-1.2.4a-sd-10.20.depot.Z> , puis décompressez la, et utilisez `swinstall` pour installer.
2. Vous aurez besoin de gcc. Téléchargez une distribution compilée à <http://gatekeep.cs.utah.edu/ftp/hpux/Gnu/gcc-2.95.2/gcc-2.95.2-sd-10.20.depot.gz> , puis décompressez la, et utilisez `swinstall` pour installer.
3. Vous aurez besoin de GNU binutils. Téléchargez une distribution compilée à

<http://hpux.connect.org.uk/ftp/hpux/Gnu/binutils-2.9.1/binutils-2.9.1-sd-10.20.depot.gz> , puis décompressez la, et utilisez swinstall pour installer.

4. Vous aurez besoin de bison. Téléchargez une distribution compilée à <http://hpux.connect.org.uk/ftp/hpux/Gnu/bison-1.28/bison-1.28-sd-10.20.depot.gz> , puis décompressez la, et utilisez swinstall pour installer.
5. Vous aurez besoin de flex. Téléchargez une distribution source sur l'un des miroirs <http://www.gnu.org>. Il se trouve dans le dossier non-gnu du site FTP. Téléchargez le fichier, gunzip , décompressez le, puis utilisez tar -xvf avec. Allez dans le nouveau dossier flex ainsi créé et exécutez la commande ./configure , puis faites un make , puis un make install .

Si vous avez des erreurs à cette étape, c'est probablement parce que gcc et les autres ne sont pas inscrites dans votre PATH. Ajoutez les. Maintenant, la partie délicate.

6. Téléchargez les sources d'Apache et de PHP.
7. Décompressez-les avec gunzip et tar -xvf . Nous devons modifier quelques fichiers avant de les compiler.
8. Premièrement, le fichier de configuration doit être modifié car il semble oublier qu'il est sur une machine HP-UX. Il y a des méthodes plus habiles, mais le plus simple et le plus efficace est d'ajouter " lt_target=hpux10.20 " à la ligne 47286 du script de configuration. script.
9. Le fichier d'Apache GuessOS doit être modifié. Sous apache_1.3.9/src/helpers , modifier la ligne 89, en remplaçant " echo "hp\${HPUXMACH}-hpux\${HPUXVER}"; exit 0 " par : " echo "hp\${HPUXMACH}-hp-hpux\${HPUXVER}"; exit 0 "
10. Il n'est pas possible d'installer PHP sous forme d'objet partagé sous HP-UX, ce qui fait que vous devez le compiler en statique. Suivez simplement les instructions de la section Apache.
11. PHP et Apache sont maintenant compilés correctement, mais Apache ne démarre pas. Vous devez créer un nouvel utilisateur Apache, par exemple www , ou apache . Puis, modifiez les lignes 252 et 253 de conf/httpd.conf pour remplacer
User nobody
Group nogroup

par vos valeurs, par exemple :

```
User www  
Group sys
```

Il n'est pas possible d'exécuter Apache avec l'utilisateur nobody sous HP-UX. A partir de ce moment là, PHP et Apache doivent fonctionner.

4.2.8 Installations sur les systèmes OpenBSD

Cette section contient les notes spécifiques à l'installation de PHP sous OpenBSD 3.6 .

4.2.8.1 Utilisation des paquets binaires

Cette méthode est la méthode recommandée pour installer PHP sur OpenBSD. C'est aussi la méthode la plus simple. Le paquet core a été séparé des modules et chacun d'entre eux peut être installé et supprimé indépendamment des autres. Les fichiers dont vous avez besoin sont sur le CD OpenBSD ou sur le site FTP.

Le paquet principal que vous devez installer est php4-core-4.3.8.tgz , qui contient le moteur de base, plus gettext et iconv). Puis, jetez un oeil aux packages de module, comme

php4-mysql-4.3.8.tgz ou php4-imap-4.3.8.tgz . Vous devez utiliser la commande phpxs pour activer et désactiver ces modules dans votre php.ini .

Exemple d'installation de PHP sous OpenBSD avec Ports

```
# pkg_add php4-core-4.3.8.tgz
# /usr/local/sbin/phpxs -s
# cp /usr/local/share/doc/php4/php.ini-recommended /var/www/conf/php.ini
  (add in mysql)
# pkg_add php4-mysql-4.3.8.tgz
# /usr/local/sbin/phpxs -a mysql
  (add in imap)
# pkg_add php4-imap-4.3.8.tgz
# /usr/local/sbin/phpxs -a imap
  (remove mysql as a test)
# pkg_delete php4-mysql-4.3.8
# /usr/local/sbin/phpxs -r mysql
  (install the PEAR libraries)
# pkg_add php4-pear-4.3.8.tgz
```

Lisez la page de manuel Unix [packages\(7\)](#) pour plus de détails sur les packages binaires d'OpenBSD.

4.2.8.2 Utilisation des ports

Vous pouvez aussi compiler PHP en utilisant [l'arbre des ports](#) . Cette méthode est recommandée aux utilisateurs expérimentés de OpenBSD. Le port PHP4 est scindé en deux sous-dossiers : core et extensions. Le dossier extensions génère les sous paquets de tous les modules PHP. Si vous souhaitez ne pas créer ces modules, vous pouvez utiliser la commande en ligne no_* FLAVOR. Par exemple, pour ne pas compiler le module imap, utilisez FLAVOR avec la valeur no_imap .

4.2.8.3 Problèmes courants

- L'installation par défaut d'Apache fonctionn dans un [contexte chroot\(2\)](#) , qui va limiter l'accès des scripts PHP au dossier /var/www . Vous devez donc créer un dossier /var/www/tmp pour que les sessions PHP soient stockées, ou bien utiliser une autre solution de sauvegarde. De plus, les sockets de bases de données doivent être placés dans ce dossier, ou bien utiliser l'interface localhost . Si vous utilisez des fonctions de réseau avec des fichiers comme /etc , par exemple /etc/resolv.conf , et /etc/services , vous devrez les rendre accessibles aussi dans /var/www/etc . Le paquet OpenBSD PEAR installe automatiquement les bons dossiers, ce qui fait que rien n'est nécessaire. Plus d'informations sur OpenBSD Apache sont disponibles sur [OpenBSD FAQ](#) .
- Le paquet OpenBSD 3.6 pour l'extension [gd](#) requiert XFree86. Si vous n'avez pas besoin des fonctionnalités de polices qui demandent X11, installez le paquet php4-gd-4.3.8-no_x11.tgz .

4.2.8.4 Versions plus anciennes

Les anciennes versions de OpenBSD utilisaient le système des FLAVORS pour compiler statiquement PHP. Comme il est trop difficile de générer des packages binaires avec cette méthode, elle est considérée comme obsolète. Vous pouvez toujours utiliser les anciennes versions stables, mais sachez qu'elles ne sont plus supportées par l'équipe d'OpenBSD. Si vous avez des commentaires sur le sujet, le responsable actuel est Anil Madhavapeddy ([avsm_arobase_openbsd_point_org](#)).

4.2.9 Installation sous Solaris

Cette section contient les notes et conseils d'installation de PHP sur les distributions Solaris .

4.2.9.1 Logiciels nécessaires

L'installation Solaris oublie généralement les compilateurs C, et leurs utilitaires. Lisez [cette FAQ](#) pour savoir pourquoi est-ce que les versions GNU de certains de ces outils sont nécessaires. Voici la liste des outils nécessaires :

- gcc (recommandé, mais d'autres compilateurs C peuvent fonctionner)
- make
- flex
- bison
- m4
- autoconf
- automake
- perl
- gzip
- tar
- GNU sed

De plus, vous devrez aussi installer (et peut être aussi compiler) toutes les bibliothèques nécessaires aux extensions comme MySQL, ORACLE, etc.

4.2.9.2 Utilisation des packages

Vous pouvez simplifier l'installation Solaris en utilisant pkgadd pour installer la plupart des composants.

4.2.10 Installation sous Gentoo

Cette section contient des notes et astuces spécifiques à l'installation de PHP sous [Gentoo Linux](#) .

4.2.10.1 Utilisation de Portage (emerge)

Au lieu de télécharger les sources de PHP et de les compiler vous-même, l'utilisation du système de paquets de Gentoo est la méthode la plus simple et la plus propre pour installer PHP. Si vous n'êtes pas familier avec la compilation de logiciels sous Linux, c'est la voie à suivre.

Si vous avez compilé votre système Gentoo jusqu'ici, vous avez déjà dû utiliser Portage. L'installation de Apache et de PHP n'est pas très différente de celle des autres utilitaires système.

La première décision à prendre est de choisir entre Apache 1.3.x et Apache 2.x. Bien que les deux puissent être utilisés avec PHP, les instructions qui suivent utiliseront Apache 1.3.x. Une autre chose à considérer est si votre arborescence locale de Portage est à jour ou non. Si vous ne l'avez pas mise à jour récemment, vous devez exécuter la commande emerge sync avant de faire quoi que ce soit d'autre. De ce fait, vous serez en mesure d'utiliser la version la plus récente d'Apache et de PHP.

Maintenant que tout est en place, vous pouvez utiliser l'exemple suivant pour installer Apache et PHP :

Exemple d'installation avec Apache 1.3 sous Gentoo

```
# emerge \<apache-2
# USE="-*" emerge php mod_php
# ebuild /var/db/pkg/dev-php/mod_php-<votre version de PHP>/mod_php-<votre version de PHP>.ebuild
# nano /etc/conf.d/apache
  Add "-D PHP4" to APACHE_OPTS

# rc-update add apache default
# /etc/init.d/apache start
```

Vous pouvez trouver plus d'informations sur emerge dans l'excellent [manuel Portage](#) issu du site web de Gentoo.

Si vous devez utiliser Apache 2, vous pouvez simplement exécuter la commande emerge apache dans l'exemple précédent.

4.2.10.2 Meilleur contrôle de la configuration

Dans la section précédente, PHP a été émergé sans aucun module d'activé. A l'heure de l'écriture de cette documentation, le seul module activé par défaut avec Portage est XML, qui est nécessaire pour [PEAR](#) . Cela n'est pas forcément ce que vous voulez et vous devriez découvrir rapidement que vous avez besoin de plus de modules, comme MySQL, gettext, GD, etc.

Lorsque vous compilez PHP depuis les sources vous-même, vous devez activer les modules en utilisant la commande configure . Avec Gentoo, vous pouvez simplement définir des flags USE qui seront passés au script de configuration automatiquement. Pour voir les tags USE à utiliser avec emerge, vous pouvez essayer :

Récupération de la liste des flags USE

```
# USE="-*" emerge -pv php

[ebuild N      ] dev-php/php-4.3.6-r1  -X -berkdb -crypt -curl -debug -doc
-fdftk -firebird -flash -freetds -gd -gd-external -gdbm -gmp -hardenedphp
-imap -informix -ipv6 -java -jpeg -kerberos -ldap -mcal -memlimit -mssql
-mysql -ncurses -nls -oci8 -odbc -pam -pdflib -png -postgres -qt -readline
-snmp -spell -ssl -tiff -truetype -xml2 -yaz  3,876 kB
```

Comme vous pouvez le voir dans l'affichage précédent, PHP admet beaucoup de flags USE. Regardez les et choisissez ce dont vous avez besoin. Si vous choisissez un flag et que vous n'avez pas la bonne bibliothèque, Portage la compilera pour vous. C'est une bonne idée d'utiliser la commande emerge -pv encore une fois pour voir ce que Portage va compiler en accord avec les flags que vous avez placés. Par exemple, si vous n'avez pas d'installation de X et que vous choisissez d'inclure X dans les flags USE, Portage compilera X en priorité par rapport à PHP, ce qui peut prendre un certain nombre d'heures.

Si vous choisissez de compiler PHP avec le support MySQL, CURL et GD, la commande devrait ressembler à ceci :

Installation de PHP avec les flags USE

```
# USE="-* curl mysql gd" emerge php mod_php
```

Tout comme dans le dernier exemple, n'oubliez pas d'émerger PHP tout comme mod_php. "php" ici signifie la version de PHP en ligne de commande, alors que "mod_php" correspond au module Apache.

4.2.10.3 Problèmes fréquents

- Si vous voyez les sources de PHP au lieu de ce que devrait produire votre script, vous avez certainement oublié d'éditer le fichier `/etc/conf.d/apache`. Apache doit être démarré avec le flag `-D PHP4`. Pour vérifier si le flag est présent, vous devriez être en mesure de le voir lorsque vous utilisez la commande `ps ax | grep apache` pendant que Apache fonctionne.
- En raison de problèmes de localisation, vous pourriez avoir des soucis lorsque plusieurs versions de PHP sont installées sur votre système. Dans ce cas, vous devez supprimer les anciennes versions manuellement en utilisant la commande `emerge unmerge mod_php-<old version>`.
- Si vous ne pouvez pas émerger PHP à cause de Java, essayez en exécutant la commande `-*` au début des flags USE comme dans l'exemple précédent.
- Si vous avez des soucis de configuration Apache et PHP, vous pouvez toujours chercher sur les [forums Gentoo](#). Essayez d'y rechercher les mots clés " Apache PHP ".

4.2.11 Installation sous Debian GNU/Linux

Cette section contient des notes et astuces spécifiques à l'installation de PHP sous [Debian GNU/Linux](#).

4.2.11.1 Utilisation de APT

Au lieu de télécharger les sources de PHP et de les compiler vous-même, l'utilisation du système de paquets de Debian est la méthode la plus simple et la plus propre pour installer PHP. Si vous n'êtes pas familier avec la compilation de logiciels sous Linux, c'est la voie à suivre.

La première décision à prendre est de choisir entre Apache 1.3.x et Apache 2.x. Les paquets PHP correspondant sont nommés respectivement `libapache-mod-php*` et `libapache2-mod-php*`. La procédure ci-dessous considérera Apache 1.3.x. A noter que, lors de la rédaction de ce manuel, aucun paquet officiel de PHP 5 n'est encore disponible. Ainsi, la procédure s'occupera de l'installation de PHP 4.

PHP est aussi disponible en version CGI ou CLI, dont les paquets sont nommés respectivement `php4-cgi` et `php4-cli`. Si vous en avez besoin, vous aurez juste à reproduire la procédure avec les bons noms de paquet. `php4-pear` est un autre paquet spécial que vous pourriez être amené à installer. Il contient une installation minimale de PEAR ainsi que l'utilitaire `pear` en ligne de commande.

Si vous avez besoin de paquets plus récents que les paquets stables issus de Debian, ou si certains modules ne sont pas présents dans la distribution officielle, vous pouvez jeter un oeil sur <http://www.apt-get.org/>. Un des résultats de la recherche sera [Dotdeb](#). Ce référentiel non officiel est entretenu par [Guillaume Plessis](#) et contient des paquets Debian des versions les plus récentes de PHP 4 et PHP 5. Pour l'utiliser, ajoutez simplement les deux lignes suivantes dans votre fichier `/etc/apt/sources.lists` et lancez la commande `apt-get update` :

Les deux lignes relatives à Dotdeb

<pre>deb http://packages.dotdeb.org stable all deb-src http://packages.dotdeb.org stable all</pre>
--

Enfin, vous devez vous assurer que votre liste de paquets est récente. Si vous ne l'avez pas mise à jour récemment, lancez la commande `apt-get update`. Ainsi, vous vous assurez d'utiliser les paquets les plus récents de Apache et PHP.

Maintenant que tout est en place, utilisez les commandes suivantes pour installer Apache et PHP :

Exemple d'installation sous Debian avec Apache 1.3

```
# apt-get install libapache-mod-php4
```

APT installera automatiquement le module PHP 4 pour Apache 1.3 ainsi que toutes ses dépendances et les activer. S'il ne vous a pas été demandé de redémarrer Apache pendant l'installation, vous devrez le faire manuellement :

Arrêter et démarrer Apache une fois que PHP 4 est installé

```
# /etc/init.d/apache stop
# /etc/init.d/apache start
```

4.2.11.2 Meilleur contrôle de la configuration

Dans la partie précédente, PHP a été installé juste avec les modules de base. Ce n'est pas forcément ce que vous voulez et vous pourriez découvrir rapidement que vous avez besoin de davantage de modules, comme MySQL, CURL, GD, etc.

Lorsque vous compilez PHP par vous même, vous devez activer les modules via la commande configure Avec APT, vous avez juste besoin d'installer des paquets supplémentaires. Ils sont tous nommés 'php4-*' (ou 'php5-*' si vous avez installé PHP 5 d'un référentiel tiers).

Obtenir la liste des paquets supplémentaires

```
# dpkg -l 'php4-*'
```

Comme vous pouvez vous en rendre compte avec le résultat de la commande précédente, les modules PHP que vous pouvez installer sont nombreux (mis à part les exceptions php4-cgi, php4-cli and php4-pear décrits précédemment). Choisissez les avec soin selon vos besoins. Si vous choisissez un module et que les bibliothèques dont il dépend ne sont pas présentes sur votre système, APT les installera pour vous.

Si vous voulez installer les modules MySQL, cURL et GD, la commande à lancer sera la suivante :

Installation des modules PHP de MySQL, cURL et GD

```
# apt-get install php4-mysql php4-curl php4-gd
```

APT ajoutera automatiquement les lignes adéquates dans vos différents php.ini (/etc/php4/apache/php.ini , /etc/php4/cgi/php.ini , etc).

Ces lignes ajoutent le support MySQL, cURL et GD au sein de PHP

```
extension=mysql.so
extension=curl.so
extension=gd.so
```

Vous aurez juste à stopper/redémarrer Apache de la même façon que précédemment.

4.2.11.3 Problèmes fréquents

- Si vous voyez la source de vos scripts PHP plutôt que le résultat attendu du script, c'est qu'APT a dû ne pas inclure /etc/apache/conf.d/php4 dans la configuration de votre Apache 1.3. Assurez vous donc que la ligne suivante est présente dans votre fichier /etc/apache/httpd.conf puis stoppez/redémarrez Apache :

Cette ligne active le support de PHP au sein d'Apache

```
# Include /etc/apache/conf.d/
```

- Si vous avez installé un module PHP supplémentaire et que ses fonctions ne sont pas disponibles dans vos scripts, assurez vous que la ligne adéquate est présente dans votre

php.ini , comme vu précédemment. Il se peut qu'APT l'omette lors de l'installation du module, à cause d'une mauvaise configuration de debconf.

4.3 Installation sur un système Mac OS X

Cette section contient les notes et conseils pour installer PHP sur un système Mac OS X. Il y a deux versions légèrement différentes de Mac OS X, Client et Serveur ; notre manuel vous guidera dans l'installation de PHP sur ces deux versions. Notez que PHP n'est pas disponible pour MacOS 9 et versions précédentes.

4.3.1 Utilisation des paquets

Il existe quelques versions pré-packagées et pré-compilées de PHP pour Mac OS X. Cela peut être utile pour mettre en place une configuration standard, mais si vous avez besoin d'accéder à des fonctionnalités spécifiques (comme un serveur sécurisé, ou un pilote de bases de données exotiques), vous aurez à compiler PHP et/ou votre serveur web vous-même. Si vous n'êtes pas familier avec la compilation et la mise en place d'applications, il est bon de vérifier si personne d'autre n'a réalisé un paquet contenant les fonctionnalités que vous désirez.

4.3.2 Compilation pour les serveurs OS X

Compilation pour les serveurs OS X

1. Téléchargez les dernières distributions d'Apache et PHP.
2. Décompressez les, et utilisez le script configure sur Apache, comme ceci.

```
./configure --exec-prefix=/usr \
--localstatedir=/var \
--mandir=/usr/share/man \
--libexecdir=/System/Library/Apache/Modules \
--iconsdir=/System/Library/Apache/Icons \
--includedir=/System/Library/Frameworks/Apple.framework/Versions/1.3/Headers \
--enable-shared=max \
--enable-module=most \
--target=apache
```

3. Si vous voulez que le compilateur fasse certaines optimisations, ajoutez cette ligne :

```
setenv OPTIM=-O2
```

4. Puis, allez dans le dossier PHP 4, et configurez PHP.

```
./configure --prefix=/usr \
--sysconfdir=/etc \
--localstatedir=/var \
--mandir=/usr/share/man \
--with-xml \
--with-apache=/src/apache_1.3.12
```

Si vous avez d'autres extensions à ajouter (MySQL, GD, etc.), assurez-vous de placer les bonnes options ici. Pour la chaîne --with-apache , ajoutez le chemin de votre distribution source apache, par exemple, /src/apache_1.3.12 .

5. Tapez make puis make install . Cela va ajouter un dossier à votre distribution Apache, sous src/modules/php4 .
6. Maintenant, reconfigurez Apache pour compiler PHP 4.

```
./configure --exec-prefix=/usr \  
--localstatedir=/var \  
--mandir=/usr/share/man \  
--libexecdir=/System/Library/Apache/Modules \  
--iconsdir=/System/Library/Apache/Icons \  
--includedir=/System/Library/Frameworks/Apache.framework/Versions/1.3/Headers \  
--enable-shared=max \  
--enable-module=most \  
--target=apache \  
--activate-module=src/modules/php4/libphp4.a
```

Vous pouvez recevoir un message qui vous dit que libmodphp4.a est trop ancien. Si c'est le cas, allez dans le dossier src/modules/php4 de votre distribution Apache, et utilisez cette commande : `ranlib libmodphp4.a` . Puis retournez à la racine de la distribution Apache, et lancez la commande configure ci-dessus. Cela aura mis la table de liens à jour. Lancez à nouveau `make` et `make install` .

7. Copiez et renommez le fichier `php.ini-dist` dans votre dossier `bin` de votre dossier PHP : `cp php.ini-dist /usr/local/bin/php.ini` ou, si vous n'avez pas de dossier local : `cp php.ini-dist /usr/bin/php.ini` .

4.3.3 Compilation pour les clients MacOs

Les instructions suivantes devraient vous aider dans l'installation du module PHP pour le serveur web Apache inclus dans MacOS X. Cette version inclut le support pour les bases de données MySQL et PostgreSQL. Ces instructions proviennent généreusement de [Marc Liyanage](#) .

Attention

Note : Soyez prudent lorsque vous faites cela, vous risquez de détruire la configuration de votre serveur web Apache!

Faites ceci pour l'installation :

1. Ouvrez une fenêtre de terminal
2. Tapez `wget http://www.diax.ch/users/liyanage/software/macosex/libphp4.so.gz` , et attendez la fin du téléchargement.
3. Tapez `gunzip libphp4.so.gz` .
4. Tapez `sudo apxs -i -a -n php4 libphp4.so`
5. Maintenant, tapez `sudo open -a TextEdit /etc/httpd/httpd.conf` . TextEdit va ouvrir le fichier de configuration de votre serveur web. Recherchez les deux lignes suivantes, à la fin du fichier (utilisez la commande de recherche) :

```
#AddType application/x-httpd-php .php  
#AddType application/x-httpd-php-source .phps
```

Supprimez les deux signes dièse (#), puis sauvez le fichier, et quittez TextEdit.

6. Finalement, tapez `sudo apachectl graceful` pour redémarrer le serveur Apache.

PHP devrait être fonctionnel. Vous pouvez le tester en créant un fichier dans votre dossier " Sites ", appelé " test.php ". Dans ce fichier, écrivez la ligne : `<?php phpinfo() ?>` .

Maintenant, ouvrez votre navigateur à l'adresse suivante : `127.0.0.1/~your_username/test.php` . Vous devriez obtenir la page d'information de `phpinfo` .

4.4 Installation sur les système Windows

Cette section est applicable à Windows 98/ME et Windows NT/2000/XP/2003. PHP devrait fonctionner sur les plates-formes 16 bits comme Windows 3.1 et parfois, on décrira les plates-formes supportées sous le nom de Win32. Windows 95 n'est plus supporté à partir de la version 4.3.0 de PHP.

Il y a deux méthodes principales pour installer PHP sous Windows : soit manuellement , soit avec l'installateur .

Si vous avez Microsoft Visual Studio, vous pouvez aussi compiler PHP à partir des sources.

Une fois que PHP est installé sur votre Windows, vous pouvez aussi ajouter diverses extensions .

Attention

Il y a beaucoup d'installateurs "tout en un" sur Internet, mais aucun n'est approuvé par Php.net car nous pensons que l'installation manuelle reste le meilleur choix pour avoir un système sécurisé et optimisé.

4.4.1 L'installateur Windows

L'installateur Windows de PHP disponible depuis les pages de <http://www.php.net/downloads.php> , installe la version CGI de PHP, et configure les serveurs web IIS, PWS, et Xitami. L'installateur n'inclut aucune extension externe supplémentaire de PHP (php_*.dll), vous les trouverez uniquement dans le paquet zippé Windows et dans les téléchargements PECL .

Note

Notez bien que bien que l'installateur soit une méthode simple pour installer PHP, il est limité dans plusieurs aspects, par exemple, la configuration automatique des extensions n'est pas prise en compte. Utilisez l'installateur n'est pas la méthode préférée pour installer PHP.

Tout d'abord, installez votre serveur HTTP favori sur votre système et assurez-vous qu'il fonctionne.

Exécutez l'installateur et suivez les instructions fournies par l'assistant. Deux types d'installation sont fournis : standard, qui utilise toutes les configurations par défaut les plus pratiques, et avancée, qui pose un maximum de questions pour paramétrer le plus finement.

L'assistant d'installation rassemble suffisamment d'informations pour configurer php.ini ainsi que certains serveurs web pour utiliser PHP. Un des serveurs web pour lequel l'installateur PHP n'effectue pas de configuration automatique est Apache, vous devez donc le configurer manuellement.

Une fois l'installation terminée, l'installateur vous informera que vous devez redémarrer. Suivez ce conseil, ou commencez à utiliser PHP immédiatement.

Attention

Gardez bien à l'esprit que cette installation de PHP n'est pas sécurisée. Si vous voulez avoir une installation sécurisée de PHP, vous devriez commencer par lire la documentation, et choisir toutes vos options avec soin. Cet installateur automatique vous permet de réaliser l'installation en un tour de main, mais n'est pas destiné à l'utilisation sur des serveurs de production.

4.4.2 Installation manuelle sous Windows

Ce guide d'installation vous aide à installer manuellement et configurer PHP avec un serveur web sous Microsoft Windows. Pour commencer, vous devrez télécharger la distribution binaire Zip sur la page <http://www.php.net/downloads.php>.

Bien qu'il existe beaucoup d'installeurs et que nous fournissons également un installeur pour Microsoft Windows, nous vous recommandons de prendre le temps de lire ceci et d'installer PHP vous-même, ce qui est la meilleure façon d'apprendre le système, et vous permettra d'installer des extensions PHP facilement lorsque vous en aurez besoin.

Note

Mise à jour d'une ancienne version de PHP

Les précédentes éditions de ce manuel vous suggéraient de déplacer les fichiers ini et les DLLS dans votre répertoire système (i.e dans le dossier C:\WINDOWS) et de ce fait, vous aviez des fichiers relatifs à PHP dans de multiples dossiers sur votre disque dur. Nous vous conseillons d'effacer tous ces fichiers (comme php.ini et les bibliothèques DLLs relatives à PHP du dossier système de Windows), avant de commencer l'installation d'une nouvelle version de PHP. Assurez-vous d'avoir effectué des sauvegardes de ces bibliothèques DLLs, sinon, vous risquez de corrompre la totalité de votre système. L'ancien fichier php.ini peut également vous aider à configurer votre nouvelle installation de PHP. Et, comme vous l'apprendrez bientôt, la méthode préférée pour installer PHP est de garder tous les fichiers relatifs à PHP dans un seul dossier et d'avoir le dossier disponible dans votre variable système PATH.

Note

Pré-requis MDAC

Si vous utilisez Microsoft **Windows 9x/NT4**, téléchargez la dernière version de Microsoft Data Access Components (MDAC) pour votre plate-forme. MDAC est disponible à <http://msdn.microsoft.com/data/>. Cette condition existe car **ODBC** est compilé dans les binaires distribués pour Windows.

Les étapes suivantes doivent être terminées sur toutes les installations avant d'exécuter une quelconque instruction spécifique au serveur.

Décompressez la distribution dans un dossier de votre choix. Si vous installez PHP 4, extrayez le fichier zippé dans C:\ car il va créer un dossier comme php-4.3.7-Win32. Si vous installez PHP 5, extrayez le fichier zippé dans C:\php car il ne va pas créer de dossier principal, comme en PHP 4. Vous pouvez choisir un autre dossier, mais soyez prudent d'éviter les espaces dans le nom du chemin au dossier (comme C:\Program Files\PHP), sinon, certains serveurs web crasheront.

La structure du dossier que vous avez extrait depuis le fichier zippé est différente pour les versions 4 et 5 de PHP et ressemble à ceci :

Structure de la distribution Windows de PHP 4

```
c:\php
|
|--cli
| |
| | -php.exe          -- Executable CLI - UNIQUEMENT pour la ligne de commande
|
+--dlls              -- DLL de support des extensions --> dossier systeme Windows
| |
| | -expat.dll
| |
| | -fdftk.dll
```

```

| |
| |-...
|
+--extensions          -- extensions DLL pour PHP
| |
| |-php_bz2.dll
| |
| |-php_cpdf.dll
| |
| |-..
|
+--mibs                -- fichiers de support de SNMP
|
+--openssl             -- fichiers de support de Openssl
|
+--pdf-related        -- fichiers de support de PDF
|
+--sapi                -- DLL SAPI
| |
| |-php4apache.dll
| |
| |-php4apache2.dll
| |
| |-..
|
+--PEAR                -- copie initiale de PEAR
|
|
|-go-pear.bat          -- script de configuration de PEAR
|
|-..
|
|-php.exe              -- exécutable CGI
|
|-..
|
|-php.ini-dist        -- paramètres par défaut du php.ini
|
|-php.ini-recommended -- paramètres recommandés du php.ini
|
|-php4ts.dll           -- DLL principale
|
|-...

```

Ou :

Structure du paquet PHP 5

```

c:\php
|
+--dev
| |
| |-php5ts.lib
| |
+--ext          -- extensions DLL pour PHP
| |
| |-php_bz2.dll
| |
| |-php_cpdf.dll
| |
| |-..
|
+--extras
| |
| +--mibs          -- fichiers de support de SNMP
| |
| +--openssl       -- fichiers de support de Openssl

```

```

| |
| +--pdf-related      -- fichiers de support de PDF
| |
| |-mime.magic
|
|--pear              -- copie initiale de PEAR
|
|
|-go-pear.bat       -- script de configuration de PEAR
|
|-fdftk.dll
|
|--..
|
|-php-cgi.exe       -- exécutable CGI
|
|-php-win.exe       -- permet d'exécuter des scripts sans ouvrir un fenêtre de prompt
|
|-php.exe           -- exécutable CLI - UNIQUEMENT pour du script en ligne de commande
|
|--..
|
|-php.ini-dist      -- paramètres par défaut du php.ini
|
|-php.ini-recommended -- paramètres recommandés du php.ini
|
|-php5activescript.dll
|
|-php5apache.dll
|
|-php5apache2.dll
|
|--..
|
|-php5ts.dll        -- DLL principale
|
|--..

```

Notez les différences et les similitudes. PHP 4 et PHP 5 ont tous les deux un exécutable CGI , un exécutable CLI et des modules serveurs, mais qu'ils sont situés dans des dossiers différents et/ou ont des noms différents. En PHP 4, les modules serveurs se trouvent dans le dossier sapi , tandis qu'ils se trouvent dans le dossier principal en PHP 5. Le support des DLLs pour les extensions de PHP sont également dans le dossier principal en PHP 5. Observez l'arborescence pour connaître l'emplacement des exécutables CGI et CLI.

Note

En PHP 4, vous devez déplacer tous les fichiers se trouvant dans les dossiers dll et sapi dans le dossier principal (e.g. C:\php).

Voici une liste de modules serveur avec la correspondance entre PHP 4 et PHP 5.

- sapi/php4activescript.dll (php5activescript.dll) - moteur ActiveScript vous permet d'intégrer PHP dans vos applications Windows.
- sapi/php4apache.dll (php5apache.dll) - module Apache 1.3.x.
- sapi/php4apache2.dll (php5apache2.dll) - module Apache 2.0.x.
- sapi/php4isapi.dll (php5isapi.dll) - module ISAPI pour les serveurs ISAPI compliant comme IIS 4.0/PWS 4.0 ou autres.
- sapi/php4nsapi.dll (php5nsapi.dll) - module serveur Sun/iPlanet/Netscape.
- sapi/php4pi3web.dll (pas d'équivalent en PHP 5) - module serveur Pi3Web.

Les modules serveurs permettent des gains de performances et quelques fonctionnalités supplémentaires par rapport à la version CGI. La version CLI est destinée à être utilisée pour les scripts en ligne de commande. Plus d'informations sur la version CLI est disponible dans le chapitre

à propos "[utilisez PHP en ligne de commande](#)".

Attention

Les modules SAPI ont été significativement améliorés dans la version 4.1, mais vous pourrez rencontrer des erreurs avec le serveur ou d'autres modules (tels ASP), dans les systèmes plus anciens.

Les binaires CGI et CLI et les modules des serveurs web requierent tous la bibliothèque php4ts.dll (php5ts.dll). Vous devez vous assurer que ce fichier peut être trouvé par votre installation de PHP. Le dossier où ce fichier sera recherché suit ces règles :

- Le même dossier depuis lequel le fichier php.exe est appelé ou, dans le cas où vous utilisez le module SAPI, le dossier du serveur web (e.g. C:\Program Files\Apache Group\Apache2\bin).
- N'importe quel dossier de votre variable d'environnement PATH .

Pour rendre le fichier php4ts.dll / php5ts.dll disponible, vous avez trois options : copiez le fichier dans le dossier système de Windwos, copiez le fichier dans le dossier du serveur web ou ajoutez le dossier PHP, C:\php à votre variable d'environnement PATH . Pour une meilleur maintenance, nous vous conseillons de suivre la dernière option et d'ajoutez le dossier C:\php à votre variable d'environnement PATH , cela rendra plus facile la mise à jour de PHP dans le futur. Lisez l' [entrée correspondante de la FAQ](#) pour avoir plus d'informations sur la façon d'ajouter votre dossier PHP à la variable d'environnement PATH .

L'étape suivante est de définir une configuration valide pour PHP, php.ini . Il y a deux fichiers ini distribués avec lepaquet zip, php.ini-dist et php.ini-recommended . Nous vous recommandons vivement d'utiliser le fichier php.ini-recommended , car nous avons optimisé les options par défaut dans ce fichier pour rendre PHP plus performant, plus sécurisé. Lisez ce document très attentivement car il contient des modifications depuis php.ini-dist qui affectent sérieusement votre configuration. Par exemple, [display_errors](#) est à off et [magic_quotes_gpc](#) est aussi à off . En complément de cette lecture, étudiez [la configuration du fichier ini](#) et définissez chacun des éléments manuellement. Si vous voulez avoir la meilleure sécurité, alors, c'est la seule façon pour vous, bien que PHP fonctionne très bien avec le fichier ini par défaut. Copiez le fichier ini de votre choix dans un dossier où PHP sera capable de le trouver et renommez le en php.ini . PHP recherche un fichier php.ini dans les endroits décrits dans la section [Le fichier de configuration](#) .

Si vous utilisez Apache 2, l'option la plus simple est d'utiliser la directive PHPIniDir (lisez la page traitant de l' [installation de PHP avec Apache 2](#)) sinon, la meilleure option est de définir la variable d'environnement PHPRC . Ce processus est expliqué dans cette [entrée de la FAQ](#) .

Note

Si vous utilisez NTFS sous Windows NT, 2000, XP ou 2003, assurez-vous que l'utilisateur faisant fonctionner le serveur web a les permissions en lecture sur votre fichier php.ini (e.g. rendez le lisible pour tout le monde).

Les étapes suivantes sont optionnelles :

- Editez votre nouveau fichier php.ini . Si vous avez prévu d'utiliser [OmniHTTPd](#) , ne suivez pas l'étape suivante. Définissez le paramètre [doc_root](#) de façon à ce qu'il pointe vers le [document root de votre serveur web](#). Par exemple :

```
doc_root = c:\inetpub\wwwroot // pour IIS/PWS
doc_root = c:\apache\htdocs // pour Apache
```

- Choisissez les extensions que vous voulez charger au démarrage de PHP. Lisez la section sur les extensions [Windows](#) , sur la manière de les configurer et celles qui sont déjà intégrées à PHP. Notez que sur les novells installations, il est préférable de faire

fonctionner PHP et de le tester avec aucune extension avant d'en activer dans votre fichier php.ini .

- Sous PWS et IIS, vous pouvez définir le paramètre de configuration browscap pour pointer vers c:\windows\system\inetsrv\browscap.ini sous Windows 9x/Me, c:\winnt\system32\inetsrv\browscap.ini sous NT/2000 et c:\windows\system32\inetsrv\browscap.ini sous XP. Pour un fichier browscap.ini à jour, lisez cette [entrée de la FAQ](#) .

PHP est maintenant installé sur votre système. L'étape suivante est le choix du serveur web et de le configurer pour y faire fonctionner PHP. Choisissez un serveur web parmi ceux supportés.

4.4.3 ActiveScript

Cette section contient des notes spécifiques à l'installation d'ActiveScript.

ActiveScript est une SAPI uniquement disponible sous Windows qui vous permet d'utiliser des scripts PHP dans l'importe quel hôte respectant ActiveScript comme Windows Scripts Host , ASP/ASP.NET , Windows Script Components ou encore Microsoft Scriptlet control .

Depuis PHP 5.0.1, ActiveScript a été déplacé dans le dépôt PECL . Vous pouvez télécharger ces bibliothèques DLL pour ces extensions PECL depuis la page [de téléchargement de PHP](#) sous sur la page <http://snaps.php.net/> .

Note

Vous devriez lire les <u>étapes d'installation du manuel</u> d'abord !
--

Après avoir installé PHP, vous devez télécharger la bibliothèque ActiveScript (php5activescript.dll) et la placer dans le dossier principale de PHP (e.g. C:\php).

Après avoir récupérer tous les fichiers nécessaires, vous devez enregistrer cette bibliothèque DLL sur votre système. Pour réaliser cela, ouvrez un prompt de commande Windows (qui se trouve dans le menu démarrer). Allez dans votre répertoire PHP en tapant quelque chose comme cd C:\php . Pour enregistrer cette bibliothèque DLL , tapez juste : regsvr32 php5activescript.dll .

Pour tester si ActiveScript fonctionne, créez un nouveau fichier, nommé test.wsf (l'extension est vraiment très importante) et tapez :

```
<job id="test">

  <script language="PHPScript">
    $WScript->Echo("Bonjour le monde !");
  </script>

</job>
```

Sauvegardez et double cliquez sur le fichier. Si vous recevez une petite fenêtre disant "Bonjour le monde !", c'est que cela fonctionne.

Note

En PHP 4, le moteur était appelé 'ActivePHP', donc, si vous utilisez PHP 4, vous devez remplacer 'PHPScript' par 'ActivePHP' dans l'exemple ci-dessus.
--

Note

ActiveScript n'utilise pas le fichier php.ini par défaut. A la place, il regardera uniquement dans le répertoire où se trouve le .exe qui l'a chargé. Vous devez créer un fichier php-activescript.ini et le
--

placer dans ce dossier si vous voulez charger des extensions, etc.

4.4.4 Installation avec les serveurs IIS/PWS

Cette section contient des notes sur l'installation de PHP avec IIS (Microsoft Internet Information Server).

Attention

En utilisant le mode CGI, votre serveur est ouvert à de possibles attaques sérieuses. Lisez attentivement notre [section sur la sécurité en mode CGI](#) pour apprendre comment vous défendre contre ces attaques.

4.4.4.1 Considérations générales pour toutes les installations de PHP avec IIS

- Tout d'abord, lisez les [instructions d'installation du manuel](#) . Ne négligez pas cette étape, elle fournit des informations essentielles sur l'installation de PHP sur Windows.
- Les utilisateurs de CGI doivent définir la directive PHP `cgi.force_redirect` à 0 dans le `php.ini` . Lisez la [faq sur cgi.force_redirect](#) qui vous apprendra des détails importants. De même, les utilisateurs de CGI peuvent vouloir définir la directive `cgi.redirect_status_env` . Lorsque vous utilisez les directives, assurez-vous qu'elles ne soient pas commentées dans le `php.ini` .
- Le CGI de PHP 4 est nommé `php.exe` tandis qu'en PHP 5, il est nommé `php-cgi.exe` . En PHP 5, `php.exe` est le CLI et non le CGI.
- Modifiez la variable d'environnement `PATH` de Windows afin d'inclure le dossier de PHP. De cette façon, les fichiers DLLs de PHP, les exécutables de PHP ainsi que le `php.ini` peuvent tous résider dans le dossier de PHP sans être dans le dossier système de Windows. Pour plus de détails, lisez la FAQ sur la [façon de définir le PATH](#) .
- L'utilisateur IIS (habituellement `IUSR_MACHINENAME`) a besoin de permission pour lire les différents fichiers et dossiers, comme le `php.ini` , `docroot` ainsi que le dossier temporaire hébergeant les sessions.
- Assurez-vous que les directives PHP `extension_dir` et `doc_root` soient correctement définies dans le `php.ini` . Ces directives dépendent du système sur lequel PHP est installé. En PHP 4, `extension_dir` vaut `extensions` tandis qu'en PHP 5, il vaut `ext` . Donc, un exemple de valeur pour `extension_dir` en PHP 5 est `"c:\php\ext"` et un exemple de valeur pour `doc_root` pour IIS est `"c:\inetpub\wwwroot"` .
- Les fichiers d'extensions DLL de PHP, comme `php_mysql.dll` et `php_curl.dll` , peuvent être trouvés dans le paquet compressé de PHP (et non dans l'installateur de PHP). En PHP 5, beaucoup d'extensions font parties de PECL et peuvent être téléchargées dans le paquet "Collection de modules PECL". Les fichiers comme `php_zip.dll` et `php_ssh2.dll` . [Téléchargez les fichiers PHP ici](#) .
- Lors de la définition de l'exécutable, la case "Vérifier que ce fichier existe" doit également être cochée. Pour un faible coût au niveau performance, IIS (ou PWS) vérifiera que le fichier de script existe et proposera l'identification avant d'appeler PHP. Cela signifie que le serveur web fournira des messages d'erreur sensiblement identique à des erreurs 404 au lieu des erreurs CGI stipulant que PHP n'a pû afficher aucune donnée.

4.4.4.2 Windows NT/200x/XP et IIS 4 ou plus récent

PHP peut être installé en tant que binaire CGI ou en tant que module SAPI. Dans tous les cas, vous devez démarrer la console d'administration Microsoft (qui doit apparaître comme 'Internet Services Manager', soit depuis le menu des options Pack de votre Windows NT 4.0 ou le menu 'Control Panel=>Administrative Tools' sous Windows 2000/XP). Faîte alors un click droit sur le noeud du serveur web (ceci doit apparaître comme 'Default Web Server'), et sélectionnez 'Properties'.

Si vous voulez utiliser le binaire CGI, suivez ce qui suit :

- Sous 'Home Directory', 'Virtual Directory', ou 'Directory', faites ce qui suit :
- Modifier les permissions d'exécution en 'Scripts only'
- Cliquez sur le bouton 'Configuration', et choisissez l'onglet 'Application Mappings'. Cliquez sur 'Add' et définissez le chemin vers l'exécutable vers le fichier CGI approprié. Un exemple de valeur pour PHP 5 : C:\php\php-cgi.exe . Ajouter .php en tant qu'extension. Laissez 'Method exclusions' vide, et cochez la case 'Script engine' checkbox. Maintenant, cliquez sur Ok plusieurs fois.
- Définissez la sécurité appropriée. (Ceci est fait dans 'Internet Service Manager'), et, si votre serveur NT utilise le système de fichiers NTFS, ajoutez le droit à l'exécution pour I_USR_ pour le dossier qui contient php.exe / php-cgi.exe .

Pour utiliser le module SAPI, faites ce qui suit :

- Si vous ne souhaitez pas faire d'identification HTTP en utilisant PHP, vous pouvez (et vous devez) ignorer cette étape. Dans les filtres ISAPI, ajoutez un nouveau filtre ISAPI. Utilisez PHP en tant que nom de filtre, et ajoutez un chemin vers les fichiers php4isapi.dll / php5isapi.dll .
- Sous 'Home Directory', 'Virtual Directory', ou 'Directory', faites ce qui suit :
- Modifier les permissions d'exécution en 'Scripts only'
- Cliquez sur le bouton "Configuration" et ajoutez une nouvelle entrée dans "Application Mappings". Cliquez sur 'Add' et définissez le chemin d'exécution vers la bibliothèque DLL ISAPI appropriée. Un exemple de valeur pour PHP 5 est : C:\php\php5isapi.dll . Ajoutez .php en tant qu'extension. Laissez 'Method exclusions' vide, et cochez la case 'Script engine'. Maintenant, cliquez que Ok plusieurs fois.
- Arrêtez totalement IIS (NET STOP iisadmin)
- Démarrez IIS (NET START w3svc)

Avec IIS 6 (2003 serveur), ouvrez le gestionnaire IIS, allez aux extensions de services web, choisissez "Add a new Web service extension", entrez-y un nom comme PHP, cliquez sur le bouton 'Add' et pour la valeur, choisissez soit le fichier ISAPI (php4isapi.dll ou php5isapi.dll), soit le fichier CGI (php.exe ou php-cgi.exe), puis cochez "Set extension status to Allowed" et validez en cliquant sur OK.

Afin d'utiliser index.php en tant que page par défaut, faites ce qui suit : Depuis l'onglet 'Documents', choisissez 'Add'. Entrez-y index.php et validez en cliquant sur OK. Ajustez l'ordre en choisissant 'Move Up' ou 'Move Down'. Ceci est similaire à la définition de 'DirectoryIndex' sous Apache.

L'étape ci-dessus doit être répétée pour chaque extension qui doit être associée aux scripts PHP. .php est le plus courant, cependant .php3 peut être requis pour certaines applications.

Si vous atteignez 100% d'utilisation du CPU après quelques minutes, désactivez l'option de configuration Cache ISAPI Application de IIS.

4.4.4.3 Windows et PWS 4

PWS 4 ne supporte pas ISAPI, uniquement PHP CGI doit être utilisé.

- Editez le fichier pws-php4cgi.reg / pws-php5cgi.reg (regardez dans le dossier SAPI pour PHP4 ou dans le dossier principal pour PHP 5) pour indiquer la localisation de votre fichier php.exe / php-cgi.exe . Les slash doivent être échappés. Par exemple :
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w3svc\parameters\Script

Map] ".php"="C:\\php\\php.exe" (modifiez en C:\\php\\php-cgi.exe si vous utilisez PHP 5). Maintenant, intégrez ce fichier de registre dans votre système ; vous devriez juste avoir à double cliquer dessus.

- Dans le gestionnaire PWS Manager, faites un clic droit sur les dossiers qui supporteront PHP, et sélectionnez "Properties". Cochez l'option "Execute" et confirmez.

4.4.4.4 Windows et PWS/IIS 3

La méthode recommandée pour configurer ces serveurs est d'utiliser le fichier INF inclus dans la distribution (pws-php4cgi.reg dans le dossier SAPI pour PHP 4 ou pws-php5cgi.reg dans le dossier principal pour PHP 5). Vous pouvez éditer ce fichier, pour vous assurer que les extensions et les dossiers d'installation de PHP sont bien ceux de votre configuration. Ou alors, vous pouvez suivre les instructions suivantes pour le faire manuellement.

Attention

Ces instructions requièrent la manipulation du fichier de registre de Windows. Une erreur peut laisser votre système dans un état instable. Nous vous recommandons vivement de sauvegarder ce fichier en lieu sûr. L'équipe de développement et les traducteurs de cette documentation ne pourront pas être tenus responsable d'un quelconque dommage qui pourrait survenir dans votre registre.

- Lancez Regedit.
- Naviguez jusqu'à : HKEY_LOCAL_MACHINE /System /CurrentControlSet /Services /W3Svc /Parameters /ScriptMap .
- Dans le menu "edit", sélectionnez : New->String Value .
- Entrez l'extension que vous voulez utiliser pour les scripts PHP. Par exemple : .php
- Double-cliquez sur la chaîne, et entrez le chemin jusqu'à php.exe dans le champ "value data". Exemple : C:\\php\\php.exe "%s" %s pour PHP 4 ou C:\\php\\php-cgi.exe "%s" %s pour PHP 5.
- Répétez ces étapes pour chaque extension que vous désirez associer à vos scripts PHP.

Les étapes suivantes n'affectent pas la configuration du serveur web, et ne s'appliquent que si vous voulez que vos scripts PHP soient exécutés lorsqu'il sont exécutés en ligne de commande (par exemple, run C:\\messcripts\\test.php) ou en double-cliquant sur l'icône. Vous pouvez ignorer ces étapes si vous préférez que vos scripts PHP s'ouvrent dans un éditeur de texte, plutôt que de les voir s'exécuter lorsque vous double-cliquez dessus.

- Naviguez jusqu'à : HKEY_CLASSES_ROOT
- Dans le menu edit, sélectionnez : New->Key .
- Donnez le nom de votre extension à la clé. Par exemple : .php
- Sélectionnez le nom de la nouvelle clé dans le panneau de droite, et double-cliquez dans "default value", puis entrez phpfile .
- Répétez ces instructions pour toutes les extensions que vous avez associé aux scripts PHP.
- Créez une autre New->Key sous HKEY_CLASSES_ROOT et nommez-la phpfile .
- Sélectionnez la nouvelle clé phpfile et dans le panneau de droite, double-cliquez dans "default value" et entrez PHP Script .
- Faites un clic droit dans phpfile et sélectionnez New->Key , appelez-le Shell .
- Faites un clic droit dans Shell et sélectionnez New->Key , appelez-le open .
- Faites un clic droit dans open et sélectionnez New->Key , appelez-le command .
- Sélectionnez la nouvelle clé command et dans le panneau de droite, faites un double-clic dans "default value", puis entrez le chemin jusqu'à php.exe . Par exemple : c:\\php\\php.exe -q %1 (n'oubliez pas le %1).
- Quittez Regedit.

- Si vous utilisez PWS sous Windows, redémarrez pour prendre en compte le nouveau registre.

Les utilisateurs de PWS et IIS 3 sont prêts à utiliser leur serveur. Avec IIS 3, vous pouvez utiliser un outil bien pratique de Steven Genusa pour configurer votre carte des scripts.

4.4.5 Installer PHP sous Microsoft Windows avec Apache 1.3.x

Cette section contient des notes et conseils spécifiques pour l'installation de PHP avec Apache 1.3.x sur les systèmes Microsoft Windows. Il y a aussi des instructions et des notes spécifiques pour Apache 2 sur une page séparée .

Note
Lisez les <u>étapes d'installation du manuel</u> d'abord !

Il y a deux méthodes pour faire fonctionner PHP avec Apache 1.3.x sous Windows. La première est d'utiliser l'exécutable CGI (php.exe pour PHP 4 et php-cgi.exe pour PHP 5), l'autre est d'utiliser les modules Apache DLL. Dans les deux cas, vous devez arrêter le serveur Apache, éditer votre fichier httpd.conf pour dire à Apache de prendre PHP en compte et redémarrer Apache.

Maintenant que le module SAPI a été rendu plus stable sous Windows, nous recommandons son usage plutôt que celui de l'exécutable CGI, car il est plus transparent et sécurisé.

Bien qu'il puisse y avoir quelques différences de configuration de PHP sous Apache, le processus reste simple et à la portée du néophyte. Reportez-vous aux documentations Apache pour plus de détails sur ces directives.

Après avoir modifié le fichier de configuration, pensez à redémarrer le serveur web, par exemple avec NET STOP APACHE suivi de NET START APACHE , si vous utilisez Apache comme service Windows, ou bien utilisez vos alias classiques.

Note
Souvenez-vous que lorsque vous ajoutez des valeurs représentants un chemin dans la configuration d'Apache sous Windows, tous les anti-slash, comme c:\repertoire\fichier.ext , doivent être convertis en slashes, comme c:/repertoire/fichier.ext .

4.4.5.1 Installation de PHP en tant que module Apache

Vous devez ajouter les lignes suivantes à votre fichier de configuration Apache httpd.conf :

PHP comme module Apache 1.3.x
Cet exemple suppose que PHP est installé dans le dossier c:\php . Ajustez le chemin si ce n'est pas le cas.
Pour PHP 4 :
<pre># A ajouter à la fin de la section LoadModule # N'oubliez pas de copier ce fichier depuis le dossier sapi ! LoadModule php4_module "C:/php/php4apache.dll" # A ajouter à la fin de la section AddModule AddModule mod_php4.c</pre>
Pour PHP 5 :

```
# A ajouter à la fin de la section LoadModule
LoadModule php5_module "C:/php/php5apache.dll"

# A ajouter à la fin de la section AddModule
AddModule mod_php5.c
```

Pour les deux :

```
# Ajoutez cette ligne dans les parenthèses conditionnelles <IfModule mod_mime.c>
AddType application/x-httpd-php .php

# Pour les fichiers de syntaxe colorisée .phps, ajoutez également
AddType application/x-httpd-php-source .phps
```

4.4.5.2 Installation comme binaire CGI

Si vous avez dézippé le paquet PHP dans le répertoire c:\php\ comme décrit dans la section sur les [étapes d'installation du manuel](#) , vous devez insérer ces lignes à votre fichier de configuration Apache pour activer le binaire CGI :

PHP et Apache 1.3.x en tant que CGI

```
ScriptAlias /php/ "c:/php/"
AddType application/x-httpd-php .php

# Pour PHP 4
Action application/x-httpd-php "/php/php.exe"

# Pour PHP 5
Action application/x-httpd-php "/php/php-cgi.exe"

# spécifiez le répertoire où se trouve php.ini
SetEnv PHPRC C:/php
```

Notez que la seconde ligne dans l'exemple ci-dessus peut être trouvée dans l'actuelle version de votre httpd.conf , mais elle est commentée. Souvenez-vous également de faire correspondre le chemin c:/php/ à votre chemin actuelle vers PHP.

Attention

En utilisant le mode CGI, votre serveur est ouvert à de possibles attaques sérieuses. Lisez attentivement notre [section sur la sécurité en mode CGI](#) pour apprendre comment vous défendre contre ces attaques.

Si vous voulez présenter la source de vos fichiers PHP avec la coloration syntaxique, il n'existe pas d'option équivalent de celle de la version module de PHP. Si vous choisissez de configurer Apache pour utiliser PHP en mode CGI, vous aurez besoin d'utiliser la fonction [highlight_file](#) . Pour réaliser cela simplement, créez un script PHP dans un fichier et ajoutez ce code : `<?php highlight_file('original_php_script.php'); ?>` .

4.4.6 Installation des serveurs Apache 2.0.x sur les systèmes Microsoft Windows

Cette section contient les notes et conseils d'installation de PHP avec le serveur Apache 2.0.x sur les systèmes Microsoft Windows. Nous avons également des notes et des instructions pour Apache 1.3.x [sur une page séparée](#) .

Note

Vous devriez lire les [étapes d'installation du manuel](#) d'abord !

Attention

Nous ne recommandons pas l'utilisation de PHP dans un environnement threadé MPM, avec Apache 2. Utilisez le mode prefork MPM à la place, ou utilisez Apache 1. Pour savoir pourquoi, lisez l'entrée de la FAQ correspondante à l' [utilisation d'Apache 2 dans un environnement threadé MPM](#) .

Il est vivement recommandé de lire la [documentation Apache](#) pour avoir une meilleure connaissance du serveur web Apache 2.0.x. Lisez également les [notes spécifiques à Windows](#) pour Apache 2.0.x avant de lire cette documentation.

Attention

Apache 2.0.x est conçu pour fonctionner sur Windows NT 4.0 et Windows 2000. Actuellement, le support des versions Windows 9x est incomplet. Apache 2.0 n'est pas prévu pour fonctionner sur ces plates-formes pour l'instant.

Téléchargez la version la plus récente de [Apache 2.0.x](#) et une version de PHP. Suivez les instructions [d'installation manuelle](#) puis revenez ici pour réaliser l'intégration de PHP et Apache.

Il y a deux méthodes pour que PHP fonctionne avec Apache 2.0.x sous Windows. La première est l'interface CGI, et l'autre est le module DLL Apache. Dans les deux cas, commencez par stopper le serveur Apache, éditez le fichier httpd.conf pour configurer Apache avec le support PHP et redémarrer Apache.

Note

Souvenez-vous que lorsque vous ajoutez des valeurs représentant un chemin dans la configuration d'Apache sous Windows, tous les anti-slash, comme c:\repertoire\fichier.ext , doivent être convertis en slashes, comme c:/repertoire/fichier.ext .

4.4.6.1 Installation de PHP en mode CGI

Vous devez insérer trois lignes à votre fichier de configuration Apache httpd.conf pour configurer le binaire CGI :

PHP et Apache 2.0.x en mode CGI

```
ScriptAlias /php/ "c:/php/"
AddType application/x-httpd-php .php

# Pour PHP 4
Action application/x-httpd-php "/php/php.exe"

# Pour PHP 5
Action application/x-httpd-php "/php/php-cgi.exe"
```

Attention

En utilisant le mode CGI, votre serveur est ouvert à de possibles attaques sérieuses. Lisez attentivement notre [section sur la sécurité en mode CGI](#) pour apprendre comment vous défendre contre ces attaques.

4.4.6.2 Installation de PHP en tant que module Apache

Vous devez insérer ces deux lignes à votre fichier de configuration Apache httpd.conf our configurer le module PHP pour Apache 2.0.x :

PHP et Apache 2.0.x en tant que module

```
# Pour PHP 4, faites quelques choses comme cela :
LoadModule php4_module "c:/php/php4apache2.dll"
# N'oubliez pas de copier le fichier php4apache2.dll depuis le dossier sapi !
AddType application/x-httpd-php .php
```

```
# Pour PHP 5, faites quelques choses comme cela :  
LoadModule php5_module "c:/php/php5apache2.dll"  
AddType application/x-httpd-php .php  
  
# Configure le chemin vers le fichier php.ini  
PHPIniDir "C:/php"
```

Note

Souvenez-vous de remplacer votre chemin actuel vers PHP par c:/php/ dans l'exemple ci-dessus. Faites attention d'utiliser soit le fichier php4apache2.dll ou php5apache2.dll dans votre directive LoadModule et **non pas** php4apache.dll ou php5apache.dll sachant que les derniers sont conçus pour fonctionner avec Apache 1.3.x .

Note

Si vous voulez utiliser la négociation sur le contenu, lisez cette [entrée de la FAQ](#) .

Attention

Ne mélangez pas votre installation avec des fichiers DLL issus **de versions différentes de PHP** . Vous avez le seul choix d'utiliser le DLL et les extensions qui correspondent avec votre version téléchargée de PHP.

4.4.7 Serveurs Sun, iPlanet et Netscape servers sur Microsoft Windows

Cette section contient les notes et détails spécifiques à l'installation de PHP sur des serveurs Sun Java System Web Server, Sun ONE web Server, Netscape et iPlanet, sous Windows.

Depuis PHP 4.3.3, vous pouvez utiliser les scripts PHP avec le [module NSAPI](#) pour [gérer des listes de dossiers et des pages d'erreurs personnalisées](#) . Des fonctions supplémentaires sont disponibles pour assurer la compatibilité avec Apache. Pour du support sur les serveurs courants, voyez la [note sur les sous-requêtes](#) .

4.4.7.1 Configuration en CGI sur les serveurs Sun, iPlanet et Netscape

Pour installer PHP en CGI, suivez ce qui suit :

- Copiez le fichier php4ts.dll dans votre dossier principal (le dossier où vous avez installé Windows)
- Faites un fichier d'association depuis la ligne de commande. Tapez les lignes suivantes :

```
assoc .php=PHPScript  
ftype PHPScript=c:\php\php.exe %1 %*
```

- Dans le serveur Netscape Enterprise Administration Server, créez un dossier shellcgi et supprimez-le aussitôt (cette opération crée 5 lignes importantes dans le fichier obj.conf et permet au serveur de gérer les scripts CGI).
- Dans le serveur Netscape Enterprise Administration Server, créez un nouveau type MIME : Category: type, Content-Type: magnus-internal/shellcgi, File Suffix:php .
- Recommencez pour chaque instance de serveur web qui devra exécuter PHP.

Plus de détails sur la configuration de PHP comme CGI sont disponibles à <http://benoit.noss.free.fr/php/install-php.html>

4.4.7.2 Configuration NSAPI sur les serveurs Sun, iPlanet et Netscape

Pour installer PHP avec l'interface NSAPI, faites ceci :

- Copiez le fichier php4ts.dll dans votre dossier systemroot (le dossier où vous avez installé windows)
- Faîtes un fichier d'association depuis la ligne de commande. Tapez les lignes suivantes :

```
assoc .php=PHPScript
ftype PHPScript=c:\php\php.exe %1 %*
```

- Dans le serveur Netscape Enterprise Administration Server, créez un nouveau type MIME : Category: type, Content-Type: magnus-internal/shellcgi, File Suffix:php .
- Editez le fichier magnus.conf (pour les serveurs >= 6) ou obj.conf (pour les serveurs < 6) et ajoutez ce qui suit : Vous devez placer ces lignes après mime types init .

```
Init fn="load-modules" funcs="php4_init,php4_execute,php4_auth_trans" shlib="c:/php/sapi/p
Init fn="php4_init" LateInit="yes" errorString="Failed to initialise PHP!" [php_ini="c:/pa
```

(PHP >= 4.3.3) Le paramètre php_ini est optionnel mais si vous le définissez, vous pourrez placer votre fichier php.ini dans le dossier de configuration de votre serveur web.

- Configurez l'objet par défaut dans le fichier obj.conf (pour les classes de serveur virtuel [Sun Web Server 6.0+], dans le fichier vserver.obj.conf) : dans la section <Object name="default"> , placez cette ligne nécessairement avant toutes les lignes 'ObjectType' et après toutes les lignes 'ObjectType' :

```
Service fn="php4_execute" type="magnus-internal/x-httpd-php" [inikey=value inikey=value ..
```

(PHP >= 4.3.3) Comme paramètres supplémentaires, vous pouvez ajouter quelques valeurs spéciales du php.ini , par exemple, vous pouvez définir un docroot="/path/to/docroot" spécifique au contexte où php4_execute est appelé, non pas "On", "Off",... (cela ne fonctionnerait pas correctement), e.g. zlib.output_compression=1 à la place de zlib.output_compression="On"

- Cela n'est nécessaire que si vous voulez configurer un dossier qui ne contiendra que vos scripts PHP (tout comme un dossier cgi-bin) :

```
<Object name="x-httpd-php">
ObjectType fn="force-type" type="magnus-internal/x-httpd-php"
Service fn=php4_execute [inikey=value inikey=value ...]
</Object>
```

Après cela, vous pouvez configurer un dossier dans l'administration du serveur et lui assigner le style x-httpd-php . Tous les fichiers si trouvant seront exécutés comme étant des scripts PHP. Cela peut être pratique pour cacher l'usage de PHP en renommant les fichiers en .html .

- Redémarrer votre serveur web pour que les modifications prennent effet.
- Faîtes cela pour chaque instance du serveur web où vous voulez exécuter PHP.

Note

Plus de détails sur la configuration de PHP comme filtre NSAPI peuvent être trouvés ici : <http://benoit.noss.free.fr/php/install-php4.html>

Note

La taille de la pile que PHP utilise dépend de la configuration du serveur web. Si vous rencontrez des crashes avec les grands scripts PHP, il est recommandé d'augmenter la taille de la pile avec la console d'administration : dans la section "MAGNUS EDITOR".

4.4.7.3 Environnement CGI et modifications recommandées du php.ini

Il est important de garder en tête que iPlanet/SunONE/Netscape est un serveur web multi-threadé. Comme toutes les requêtes se situent dans le même contexte (c'est le contexte sur serveur web), et

que ce contexte est unique. Si vous voulez accéder à des variables comme `PATH_INFO`, `HTTP_HOST` etc. il n'est pas recommandé d'y accéder à la manière de PHP 3.x, avec la fonction `getenv` ou une autre méthode (`register_globals`, `$_ENV`). De cette manière, vous n'aurez que des valeurs d'environnement du serveur, et non pas des valeurs correctes pour le CGI.

<p>Note</p> <p>Pourquoi est-ce que les variables CGI sont invalides ?</p> <p>C'est lié au fait que le processus du serveur web est lancé par l'administrateur du serveur, qui utilise le script de lancement au démarrage. En fait, il aurait fallu que vous lanciez vous-même le processus. C'est pour cela que l'environnement du serveur web contient des variables d'environnement CGI. Vous pouvez vérifier cela en lançant le serveur web depuis un autre endroit que l'administrateur du serveur : utilisez la ligne de commande Unix en tant que root : vous verrez alors qu'il n'y a pas de variables d'environnement.</p>

Changez simplement vos scripts pour lire les variables CGI, en utilisant le tableau superglobal `$_SERVER`. Si vous avez d'autres scripts qui utilisent encore `$HTTP_HOST` et compagnie, il est recommandé d'activer l'option `register_globals` dans le `php.ini` et de changer l'ordre des variables. **IMPORTANT** : supprimez le "E" dans cette option, car vous n'en avez pas besoin pour cet environnement.

```
variables_order = "GPCS"
register_globals = On
```

4.4.7.4 Utilisation particulière pour les pages d'erreurs ou les listages spécifiques de dossier (PHP >= 4.3.3)

Vous pouvez utiliser PHP pour générer des pages d'erreurs de type "404 Not Found" ou apparantée. Ajoutez la ligne suivante dans le fichier `obj.conf` pour chaque page d'erreur que vous souhaitez remplacer :

```
Error fn="php4_execute" code=XXX script="/path/to/script.php" [inkey=value inkey=value...]
```

où XXX est le code d'erreur HTTP. Effacez toute autre directive `Error` qui pourrait interférer avec la votre. Si vous voulez utiliser une page pour toutes les erreurs qui existent, laissez le paramètre `code` vide. Votre script peut obtenir le code de statut HTTP dans la variable `$_SERVER['ERROR_TYPE']`.

Une autre possibilité est de générer une liste de dossier personnalisée. Créez simplement un script PHP qui affiche le contenu du dossier, et remplacez la ligne `Service` par défaut par `type="magnus-internal/directory"` dans `obj.conf` avec ceci :

```
Service fn="php4_execute" type="magnus-internal/directory" script="/path/to/script.php" [inkey=v
```

Pour ces deux points, l'URI originale et l'URI traduite sont dans les variables `$_SERVER['PATH_INFO']` et `$_SERVER['PATH_TRANSLATED']`.

4.4.7.5 Note au sujet de `nsapi_virtual` et des requêtes (PHP >= 4.3.3)

Le module NSAPI supporte désormais la fonction `nsapi_virtual` (alias : `virtual`), pour réaliser des sous requêtes au serveur web, et inclure le résultat dans une page. Le problème est que cette fonction utilise une fonctionnalité non documentée de la bibliothèque NSAPI.

Sous Unix, ce n'est pas un problème, car le module va automatiquement rechercher les fonctions nécessaires, et les utiliser si elles sont disponibles. Sinon, `nsapi_virtual` sera désactivée.

Sous Windows, des limitations dans la gestion des DLL impose l'utilisation de la plus récente bibliothèque `ns-httpdXX.dll`. Cela a été testé pour les serveurs jusqu'à la version 6.0. Si une nouvelle version de SunONE server est utilisée, la détection échoue, et `nsapi_virtual` est désactivée.

Dans ce cas, essayez ceci : ajoutez le paramètre suivant à `php4_init` dans `magnus.conf` / `obj.conf` :

```
Init fn=php4_init ... server_lib="ns-httpdXX.dll"
```

où `XX` est le numéro correct de la version de la DLL. Pour la connaître, regardez dans le `server-root` pour connaître le nom correct de la DLL. La DLL la plus grande en taille est la bonne.

Vous pouvez vérifier le status en utilisant la fonction `phpinfo`.

Note
Soyez prévenu : le support de <code>nsapi_virtual</code> est expérimental.

4.4.8 Installation pour les serveurs OmniHTTPd

Cette section contient les notes et conseils pour installer PHP sur un serveur `OmniHTTPd` sous Windows.

Note
Vous devriez lire les étapes d'installation du manuel d'abord !
Attention
En utilisant le mode CGI, votre serveur est ouvert à de possibles attaques sérieuses. Lisez attentivement notre section sur la sécurité en mode CGI pour apprendre comment vous défendre contre ces attaques.

Vous devez réaliser les étapes suivantes pour configurer PHP sur votre serveur `OmniHTTPd`. C'est une configuration en exécutable CGI. SAPI est supporté par `OmniHTTPd`, mais des tests ont prouvé que ce n'est pas une solution stable.

Note
Important pour les utilisateurs CGI
Lisez la FAQ sur <code>cgi.force_redirect</code> pour plus de détails. Cette directive doit prendre la valeur de 0.

1. Installation du serveur `OmniHTTPd`.
2. Faites un clic droit sur l'icône bleue d'`OmniHTTPd` sur le système, et sélectionnez `Propriétés`
3. Cliquez sur `Web Server Global Settings`
4. Dans l'onglet 'External', entrez : `virtual = .php | actual = c:\php\php.exe` (utilisez `php-cgi.exe` si vous installez PHP 5), et utilisez le bouton "Add" (ajout).
5. Dans l'onglet `Mime`, entrez : `virtual = wwwserver/stdcgi | actual = .php`, et utilisez le bouton "Add" (ajout).
6. Cliquez sur `OK`

Répétez les étapes de 2 à 6 pour chaque extension que vous voulez associer à PHP.

Note

Certains packages OmniHTTPd sont fournis avec le support de PHP. Vous pouvez le choisir au moment de l'installation, et décocher le composant PHP. Nous recommandons d'utiliser les dernières versions de PHP. Certains serveurs OmniHTTPd sont livrés avec des versions bêta de PHP, et il vaut donc mieux éviter de les installer, et choisir une autre version, stable. Si le serveur est déjà sur votre machine, utilisez le bouton "Replace" (remplacer) dans les étapes 4 et 5 pour configurer un nouveau PHP.

4.4.9 Sambar Server on Microsoft Windows

Cette section contient les notes et conseils d'installation de PHP sur les serveurs Sambar, sur Windows.

Note

Vous devriez lire les étapes d'installation du manuel d'abord !

Cette liste décrit comment configurer le module ISAPI avec le serveur Sambar sous Windows.

- Trouvez le fichier appelé mappings.ini (dans le dossier de configuration), dans le dossier d'installation de Sambar.
- Ouvrez mappings.ini et ajoutez la ligne suivante, sous la section [ISAPI] :

Configuration ISAPI de Sambar

```
#pour PHP 4
*.php = c:\php\php4isapi.dll

#pour PHP 5
*.php = c:\php\php5isapi.dll
```

(Cette ligne suppose que PHP a été installé dans le dossier c:\php).

- Maintenant, redémarrez le serveur Sambar pour que les modifications prennent effet.

4.4.10 Installation Xitami sur Microsoft Windows

Cette section contient les conseils d'installation spécifiques à Xitami sur Microsoft Windows.

Note

Vous devriez lire les étapes d'installation du manuel d'abord !

Cette liste décrit comment installer PHP comme CGI exécutable avec Xitami sous Windows.

Note

Important pour les utilisateurs de CGI

Lisez la FAQ sur cgi.force_redirect pour d'important détails. Cette directive doit être configurée à 0. Si vous voulez utiliser \$_SERVER['PHP_SELF'], vous devez activer la directive cgi.fix_pathinfo.

Attention

En utilisant le mode CGI, votre serveur est ouvert à de possibles attaques sérieuses. Lisez attentivement notre section sur la sécurité en mode CGI pour apprendre comment vous défendre contre ces attaques.

- Assurez-vous que le serveur web fonctionne, et allez dans la console d'administration du serveur (généralement http://127.0.0.1/admin), puis cliquez sur "Configuration".
- Naviguez dans les Filters, et ajoutez l'extension que vous souhaitez (souvent .php) dans le champ File extensions.
- Dans la commande Filter, ajoutez le nom et le chemin de votre exécutable PHP i.e

C:\php\php.exe pour PHP 4 ou C:\php\php-cgi.exe pour PHP 5.

- Cliquez sur le bouton Save .
- Redémarrez le serveur pour prendre en compte les modifications.

4.4.11 Compilation des sources

Ce chapitre va vous apprendre à compiler PHP depuis les sources sous Windows, en utilisant les utilitaires Microsoft. Pour compiler PHP avec Cygwin, référez-vous à [Installation sous UNIX](#) .

4.4.11.1 Pré-requis

Pour compiler et construire PHP, vous devez avoir un environnement de développement Microsoft. Microsoft Visual C++ 6.0 est recommandé, bien que les versions .NET (soit le Toolkit libre Microsoft Visual C++ ou le Toolkit commercial Microsoft Visual C++ .NET) doivent également fonctionner. Pour extraire les fichiers téléchargés, vous devez également avoir un utilitaire de décompression ZIP. Windows XP et supérieur inclut cette fonctionnalité en natif.

Avant de commencer, vous devez télécharger :

- les utilitaires de compilation win32 depuis le site PHP sur <http://www.php.net/extra/win32build.zip> .
- le code source pour l'utilitaire de résolution de nom DN utilisé par PHP sur http://www.php.net/extra/bindlib_w32.zip . C'est un remplacement à la bibliothèque resolv.lib inclut dans win32build.zip .
- Si vous avez prévu de compiler PHP en tant que module Apache, vous devez également avoir les [sources d'Apache](#) .

Finalement, vous avez besoin des sources de PHP elles-mêmes. Vous pouvez récupérer la dernière version de développement en utilisant le [CVS anonyme](#) , un [snapshot](#) ou la version compressée la plus récente des [sources](#) .

4.4.11.2 Compiler tout ensemble

Après avoir téléchargés les paquets requis, vous devez les extraire dans un endroit approprié :

- Créez un dossier de travail où vous y placerez tous les fichiers décompressés, e.g : C:\work .
- Créez le dossier win32build dans votre dossier de travail (C:\work) et décompressez le fichier win32build.zip dans ce dossier.
- Créez un dossier bindlib_w32 dans votre dossier de travail (C:\work) et décompressez le fichier bindlib_w32.zip dans ce dossier.
- Décompressez les sources PHP téléchargées dans votre dossier de travail (C:\work) .
- Construisez les bibliothèques dont vous avez besoin (ou téléchargez les binaires s'ils sont disponibles) et placez les en-têtes et les bibliothèques respectivement dans les dossiers C:\work\win32build\include et C:\work\win32build\lib .

Si vous avez correctement suivis les étapes précédentes, vous devez obtenir la structure suivante :

```
+--C:\work
|
| |
| | +-bindlib_w32
| |
| | |
```

```

| | +--arpa
| | |
| | +--conf
| | |
| | +--...
| |
| +--php-5.x.x
| | |
| | +--build
| | |
| | +--...
| | |
| | +--win32
| | |
| | +--...
| |
| +--win32build
| | |
| | +--bin
| | |
| | +--include
| | |
| | +--lib

```

Si vous utilisez [Cygwin](#) , vous devez également créer les dossiers C:\usr\local\lib , puis copier le fichier bison.simple depuis C:\work\win32build\bin dans le dossier C:\usr\local\lib .

Note

Si vous voulez utiliser PEAR ainsi que son installateur en ligne de commande, CLI-SAPI est nécessaire. Pour plus d'informations sur PEAR et son installateur, lisez la documentation sur le site web de [PEAR](#) .

4.4.11.3 Construction de resolv.lib

Vous devez construire la bibliothèque resolv.lib . Décidez si oui ou non vous voulez de disponible les symboles de débogage (bindlib - Win32 Debug) ou pas (bindlib - Win32 Release), mais souvenez-vous de ce choix, car vous devrez construire PHP de la même façon, sinon vous aurez des erreurs de liaisons. Construisez la configuration appropriée :

- Pour les utilisateurs de GUI, lancez VC++ en double-cliquant sur le fichier C:\work\bindlib_w32\bindlib.dsw . Puis, sélectionnez "Build=>Rebuild All".
- Pour les utilisateurs de lignes de commande, assurez-vous d'avoir soit les variables d'environnements C++ de disponibles, soit d'avoir exécuté vcvars.bat , puis exécutez une des commandes suivantes :
 - ◆ msdev bindlib.dsp /MAKE "bindlib - Win32 Debug"
 - ◆ msdev bindlib.dsp /MAKE "bindlib - Win32 Release"

À ce point, vous devez avoir une bibliothèque utilisable de resolv.lib dans le dossier C:\work\bindlib_w32\Debug ou dans le sous-dossier Release . Copiez ce fichier dans votre dossier C:\work\win32build\lib en écrasant le fichier portant le même nom.

4.4.11.4 Construction de PHP en utilisant le nouveau système de construction [PHP >=5 uniquement]

Ce chapitre explique comment compiler PHP >=5 en utilisant le nouveau système de construction, basé sur CLI et extrêmement similaire au système de construction Unix pour PHP.

Note

Ce système de construction n'est pas disponible pour PHP 4. Référez-vous plutôt à Compilation de PHP en utilisant les fichiers DSW [PHP 4] .
--

Avant de commencer, assurez-vous d'avoir lu [Compiler tout ensemble](#) et d'avoir construit toutes les bibliothèques nécessaires, comme [Libxml](#) ou [ICU](#) (nécessaire pour PHP >= 6).

D'abord, vous devez ouvrir un prompt de commande Visual Studio, qui doit être disponible avec le menu Démarrer. Un prompt de commande classique ne fonctionnera pas, car il n'aura probablement pas les variables d'environnements nécessaires. Puis, tapez quelque chose comme `cd C:\work\php-5.x.x` pour entrer dans le dossier contenant les sources de PHP. Maintenant, vous êtes prêt à configurer PHP.

La seconde étape consiste à exécuter le fichier `buildconf` afin de lancer le script de configuration, en scannant le dossier à la recherche des fichiers `config.w32` . Par défaut, cette commande recherchera également dans les dossiers suivants : `pecl`; `..\pecl`; `pecl\rpc`; `..\pecl\rpc` . Depuis PHP 5.1.0, vous pouvez changer ce comportement en utilisant l'argument `--add-modules-dir` (e.g. `cscript /nologo win32/build/buildconf.js --add-modules-dir=../php-gtk2 --add-modules-dir=../pecl`).

La troisième étape consiste en la configuration. Pour voir la liste des options de configuration, tapez la commande `cscript /nologo configure.js --help` .Après avoir choisies les options à activer ou non, tapez quelque chose comme : `cscript /nologo configure.js --disable-foo --enable-fun-ext` . En utilisant l'argument `--enable-foo=shared` , vous tenterez de construire l'extension 'foo' en tant que module partagé, dynamiquement chargeable.

La dernière étape est la compilation. Pour ce faire, entrez simplement la commande `nmake` . Les fichiers générés (e.g. `.exe` et `.dll`) seront placés soit dans le dossier `Release_TS` , soit dans le dossier `Debug_TS` (si vous avez compilé avec le Thread safety), ou sinon, dans le dossier `Release` ou le dossier `Debug` .

Optionnellement, vous pouvez vouloir exécuter une suite de test PHP en tapant la commande `nmake test` . Si vous ne voulez effectuer qu'un test précis, vous devez utiliser la variable 'TESTS' (e.g. `nmake /D TESTS=ext/sqlite/tests test` - exécutera que les tests `sqlite`). Pour effacer les fichiers créés pendant la compilation, vous pouvez utiliser la commande `nmake clean` .

Une option de configuration vraiment utile pour la construction de snapshots est l'option `--enable-snapshot-build` , qui ajoute un nouveau mode de compilation (`nmake build-snap`). Celui-ci tente de compiler chaque extension de disponible (en tant que partagé par défaut) en ignorant les erreurs de compilation dans les extensions individuelles ou SAPI.

4.4.11.5 Compilation de PHP en utilisant les fichiers DSW [PHP 4]

La compilation de PHP en utilisant les fichiers DSW n'est plus supportée depuis PHP 5 car un [système plus flexible a été mis à disposition](#) . Malgré tout, vous pouvez continuer à l'utiliser mais gardez à l'esprit que ce système n'est plus maintenu régulièrement, vous pourriez donc rencontrer des problèmes lors de la compilation. Cependant, pour compiler PHP 4 sous Windows, il n'y a que ce système de disponible.

4.4.11.5.1 Configuration de MVC ++

La première étape consiste à la configuration de MVC++ pour la préparation à la compilation. Lancez Microsoft Visual C++ et depuis le menu, sélectionnez "Tools => Options". Dans la fenêtre de dialogue, sélectionnez la tabulation des dossiers. Séquentiellement, modifier la liste en "Executables", "Includes", et "Library files". Votre entrée devrait ressembler à ceci :

- Executable files: C:\work\win32build\bin , Cygwin users: C:\cygwin\bin
- Include files: C:\work\win32build\include
- Library files: C:\work\win32build\lib

4.4.11.5.2 Compilation

La meilleure façon de commencer est de compiler la version CGI :

- Pour les utilisateurs d'interface, lancez VC++, sélectionnez "File => Open Workspace" puis, sélectionnez C:\work\php-4.x.x\win32\php4ts.dsw . Alors, sélectionnez "Build=>Set Active Configuration" et sélectionnez la configuration désirée, soit php4ts - Win32 Debug_TS , soit php4ts - Win32 Release_TS . Et pour finir, sélectionnez "Build=>Rebuild All".
- Pour les utilisateurs de la ligne de commande, assurez-vous d'avoir de disponible les variables d'environnements C++ ou d'avoir exécuté le fichier vcvars.bat , puis, exécutez une des commandes suivantes depuis le répertoire C:\work\php-4.x.x\win32 :
 - ◆ msdev php4ts.dsp /MAKE "php4ts - Win32 Debug_TS"
 - ◆ msdev php4ts.dsp /MAKE "php4ts - Win32 Release_TS"
 - ◆ À ce stade, vous devez avoir un binaire php.exe fonctionnel dans le dossier C:\work\php-4.x.x\Debug_TS ou dans le sous-dossier Release_TS .

Il est possible de faire quelques personnalisations sur le processus de compilation en éditant le fichier main/config.win32.h . Par exemple, vous pouvez modifier le dossier par défaut contenant le php.ini , les extensions internes et le dossier par défaut de vos extensions.

Puis, vous pourriez vouloir compiler la version CLI qui est destiné à utiliser PHP depuis la ligne de commande . Les étapes sont les mêmes que pour la compilation de la version CGI, mise à part que vous devez sélectionner le fichier de projet php4ts_cli - Win32 Debug_TS ou php4ts_cli - Win32 Release_TS . Après une compilation réussie, vous devriez trouver le fichier php.exe dans le dossier Release_TS\cli\ ou dans le dossier Debug_TS\cli\ .

Pour compiler le module SAPI (php4isapi.dll) pour intégrer PHP avec Microsoft IIS, définissez votre configuration active à php4isapi-whatever-config et compilez la bibliothèque dll désirée.

4.4.12 Installation des extensions sous Windows

Après avoir installé PHP et un serveur web sous Windows, vous devriez probablement vouloir installer quelques extensions pour avoir des fonctionnalités supplémentaires. Vous pouvez choisir quelles extensions seront chargées lors du démarrage de PHP en modifiant votre php.ini . Vous pouvez également en charger dynamiquement dans vos scripts à l'aide de la fonction dl .

Les bibliothèques DLLs pour les extensions PHP sont préfixées par php_ .

Beaucoup d'extensions sont **incluses** dans la version pour Windows de PHP. Cela signifie que les bibliothèques DLL additionnelles et la directive extension ne sont **pas** utilisées pour charger ces extensions. La table des extensions PHP pour Windows liste les extensions qui requièrent des bibliothèques DLL additionnelles PHP. Voici une liste d'extensions internes :

En PHP 4 (mise à jour : PHP 4.3.11) : [BCMath](#) , [Calendar](#) , [COM](#) , [Ctype](#) , [FTP](#) , [MySQL](#) , [ODBC](#) , [Overload](#) , [PCRE](#) , [Session](#) , [Tokenizer](#) , [WDDX](#) , [XML](#) et [Zlib](#)

En PHP 5 (mise à jour : PHP 5.0.4), les changements suivants existent. En interne : [DOM](#) , [LibXML](#) , [Iconv](#) , [SimpleXML](#) , [SPL](#) et [SQLite](#) . Les suivants ne sont plus intégrés : [MySQL](#) and [Overload](#) .

Le dossier par défaut dans lequel PHP cherche des extensions est c:\php4\extensions en PHP 4 et c:\php5 en PHP 5. Pour changer ce comportement pour refléter votre installation de PHP, éditez votre fichier php.ini :

- Vous devriez pouvoir changer le paramètre `extension_dir` pour pointer vers le dossier contenant vos extensions ou l'endroit où vous avez placé vos fichiers `php_*.dll` . Faites attention à ne pas oublier le dernier anti-slashe. Par exemple :

```
extension_dir = c:/php/extensions/
```

- Pour activer ces extensions dans votre `php.ini` , vous devez décommenter les lignes `extension=php_*.dll` dans votre `php.ini` . Cela se fait en effaçant le point virgule (" ; ") du début de la ligne que vous voulez activer.

Activer l'extension [Bzip2](#) pour PHP-Windows

```
// changez la ligne suivante :
;extension=php_bz2.dll

// En :
extension=php_bz2.dll
```

- Quelques extensions ont besoin de bibliothèques DLLs supplémentaire pour fonctionner. La plus part d'entre elles peut être trouvées dans le paquet de votre distribution de PHP, dans le dossier `c:\php\dlls\` en PHP 4 ou dans le dossier principal en PHP 5 mais quelques autres, comme Oracle (`php_oci8.dll`) requiert des DLLs qui ne sont pas fournies avec votre distribution de PHP. Si vous installez PHP 4, copiez les bibliothèques DLLs depuis le dossier `C:\php\dlls` vers le dossier principal `C:\php` . N'oubliez pas d'inclure le dossier `C:\php` dans la variable d'environnement `PATH` (ce processus est expliqué dans une [entrée de la FAQ](#)).
- Quelques unes de ces bibliothèques ne sont pas incluses dans la distribution de PHP. Lisez la documentations de chaque extension pour plus de détails. Lisez également la section du manuel nommé [Installation d'extensions PECL](#) pour plus de détails sur PECL . Un nombre toujours plus important d'extensions PHP se trouve dans PECL , et ces extensions nécessitent un [téléchargement séparé](#) .

Note

Si vous utilisez PHP en tant que module d'un serveur web, pensez à redémarrer votre serveur web pour charger les modifications apportées au fichier `php.ini` .

La table suivante décrit quelques extensions disponibles requirants des bibliothèques DLLs supplémentaires.

Extension	Description	Notes
<code>php_bz2.dll</code>	bzip2 : fonctions de compression	Non
<code>php_calendar.dll</code>	Calendar : fonctions de conversions	Intégré à PHP depuis la version 4.0.3
<code>php_cpdf.dll</code>	Fonctions ClibPDF	Non
<code>php_crack.dll</code>	Fonctions Crack	None
<code>php_ctype.dll</code>	Famille de fonctions ctype	Intégré à PHP depuis la version 4.3.0
<code>php_curl.dll</code>	Fonctions de bibliothèque client CURL	Requiert : <code>libeay32.dll</code> , <code>ssleay32.dll</code> (intégré)

php_cybercash.dll	Fonctions de paiement <u>Cybercash</u>	PHP <= 4.2.0
php_db.dll	Fonctions <u>DBM</u>	Déprécier. Utilisez plutôt DBA (php_dba.dll)
php_dba.dll	<u>DBA</u> : DataBase (dbm-style) Fonctions d'abstraction	Non
php_dbase.dll	Fonctions <u>dBase</u>	Non
php_dbx.dll	Fonctions <u>dbx</u>	
php_domxml.dll	Fonctions <u>DOM XML</u>	PHP <= 4.2.0 requiert : libxml2.dll (intégré) PHP >= 4.3.0 requiert : iconv.dll (intégré)
php_dotnet.dll	Fonctions <u>.NET</u>	PHP <= 4.1.1
php_exif.dll	En-tête de <u>lecture EXIF</u> pour JPEG	php_mbstring.dll. Attention, php_exif.dll doit être chargé après php_mbstring.dll dans le php.ini .
php_fbsql.dll	Fonctions <u>FrontBase</u>	PHP <= 4.2.0
php_fdf.dll	<u>FDF</u> : Fonctions Forms Data Format.	Requiert : fdftk.dll (intégré)
php_filepro.dll	Fonctions <u>filePro</u>	Accès en lecture seul
php_ftp.dll	Fonctions <u>FTP</u>	Intégré à PHP depuis la version 4.0.3
php_gd.dll	<u>GD</u> : Bibliothèque de fonctions image	Supprimer en PHP 4.3.2. Notez que les fonctions sur les couleurs vraies ne sont pas disponibles en GD1 ; utilisez plutôt php_gd2.dll .
php_gd2.dll	<u>GD</u> : Bibliothèque de fonctions image	GD2
php_gettext.dll	Fonctions <u>Gettext</u>	PHP <= 4.2.0 requiert gnu_gettext.dll (intégré), PHP >= 4.2.3 requiert libintl-1.dll , iconv.dll (intégré).
php_hyperwave.dll	Fonctions <u>HyperWave</u>	Non
php_iconv.dll	<u>ICONV</u> : Conversion de jeux de caractères	Requiert : iconv-1.3.dll (intégré), PHP >=4.2.1 iconv.dll
php_ifx.dll	Fonctions <u>Informix</u>	Requiert : bibliothèque Informix
php_iisfunc.dll	Fonctions d'administration <u>IIS</u>	Non
php_imap.dll	<u>IMAP</u> : Fonctions POP3 et NNTP	Non
php_ingres.dll	Fonctions <u>Ingres II</u>	Requiert : bibliothèque Ingres II
php_interbase.dll	Fonctions <u>InterBase</u>	Requiert : gds32.dll (intégré)
php_java.dll	Fonctions <u>Java</u>	PHP <= 4.0.6 requiert : jvm.dll (intégré)
php_ldap.dll	Fonctions <u>LDAP</u>	PHP <= 4.2.0 requiert libsasl.dll (intégré), PHP >= 4.3.0 requiert libeay32.dll , ssleay32.dll (intégré)
php_mbstring.dll	Fonctions <u>Multi-Byte String</u>	Non
php_mcrypt.dll	Fonctions <u>Mcrypt Encryption</u>	Requiert : libmcrypt.dll
php_mhash.dll	Fonctions <u>Mhash</u>	PHP >= 4.3.0 requiert : libmhash.dll (intégré)
php_mime_magic.dll	Fonctions <u>Mimetype</u>	Requiert : magic.mime (intégré)
php_ming.dll	Fonctions <u>Ming</u> pour Flash	Non
php_msql.dll	Fonctions <u>mSQL</u>	Requiert : msql.dll (intégré)
php_mssql.dll	Fonctions <u>MSSQL</u>	Requiert : ntwdblib.dll (intégré)

php_mysql.dll	<u>MySQL</u> functions	PHP >= 5.0.0, requires libmysql.dll (intégré)
php_mysqli.dll	<u>MySQLi</u> functions	PHP >= 5.0.0, requires libmysql.dll (libmysql.dll en PHP <=5.0.2) (intégré)
php_oci8.dll	Fonctions <u>Oracle 8</u>	Requiert : bibliothèque cliente Oracle 8.1+
php_openssl.dll	Fonctions <u>OpenSSL</u>	Requiert : libeay32.dll (intégré)
php_oracle.dll	Fonctions <u>Oracle</u>	Requiert : bibliothèque cliente Oracle 7
php_overload.dll	Fonctions <u>Object overloading</u>	Intégré à PHP depuis la version 4.3.0
php_pdf.dll	Fonctions <u>PDF</u>	Non
php_pgsql.dll	Fonctions <u>PostgreSQL</u>	Non
php_printer.dll	Fonctions <u>Printer</u>	Non
php_shmop.dll	Fonctions de <u>partage de mémoire</u>	Non
php_snmp.dll	Fonctions <u>SNMP</u>	NT seulement !
php_soap.dll	<u>SOAP</u> functions	PHP >= 5.0.0
php_sockets.dll	Fonctions <u>Socket</u>	Non
php_sybase_ct.dll	Fonctions <u>Sybase</u>	Requiert : bibliothèque cliente Sybase
php_tidy.dll	<u>Tidy</u> functions	PHP >= 5.0.0
php_tokenizer.dll	Fonctions <u>Tokenizer</u>	Intégré à PHP depuis la version 4.3.0
php_w32api.dll	Fonctions <u>W32api</u>	Non
php_xmlrpc.dll	Fonctions <u>XML-RPC</u>	PHP >= 4.2.1 requiert : iconv.dll (intégré)
php_xslt.dll	Fonctions <u>XSLT</u>	PHP <= 4.2.0 requiert sablot.dll , expat.dll (intégré). PHP >= 4.2.1 requiert sablot.dll , expat.dll et iconv.dll (intégré).
php_yaz.dll	Fonctions <u>YAZ</u>	Requiert : yaz.dll (intégré)
php_zip.dll	Fonctions <u>Zip File</u>	Accès en lecture seule
php_zlib.dll	Fonctions de compression <u>ZLib</u>	Intégré à PHP depuis la version 4.3.0

4.5 Installation d'extensions PECL

4.5.1 Introduction aux installations PECL

PECL est un dépôt d'extensions PHP qui vous sont disponibles via le système de paquet PEAR . Cette section du manuel vous guide dans l'obtention et l'installation d'extensions PECL.

Ces instructions supposent que /your/phpsrcdir/ est le chemin jusqu'à les sources de la distribution PHP, et extname est le nom de votre extension PECL : adaptez les commandes qui suivent à votre situation. Ces instructions supposent aussi que vous êtes familiers avec l'utilisation des commandes pear .

Pour être utilisée, une extension partagée doit être compilée, installée et chargée. Les méthodes décrites ci-dessous vous fournissent diverses instructions sur la façon de compiler et d'installer des extensions mais ne les chargent pas automatiquement. Les extensions peuvent être chargées en ajoutant une directive d' extension au fichier php.ini ou à l'aide de la fonction dl .

Lorsque vous compilez des modules PHP, il est important d'avoir les outils dans leurs versions appropriées, tels que autoconf, automake, libtool, etc. Voyez les Instructions pour le CVS anonyme

, afin de connaître les utilitaires nécessaires, et leurs versions.

4.5.2 Télécharger des extensions PECL

Il existe plusieurs méthodes pour télécharger des extensions PECL :

- <http://pecl.php.net>

Le site web de PECL contient diverses informations sur les différentes extensions offertes par l'équipe de développement de PHP. Vous pourrez consulter les modifications entre les versions, les notes de versions, ce qui est requis pour faire fonctionner l'extension ainsi que d'autres détails similaires.

- `pear download extname`

Les extensions PECL listées sur le site web de PECL sont disponibles et peuvent être téléchargées et installées en utilisant la commande pear. La version spécifique de l'extension peut également être spécifiée.

- CVS

La plupart des fichiers PECL sont conservés dans le serveur CVS. Une interface Web est disponible à <http://cvs.php.net/pecl/>. Pour télécharger directement depuis CVS, suivez la séquence d'instructions ci-dessous. Notez que **phpfi** est le mot de passe de l'utilisateur **cvsread** :

```
$ cvs -d:pserver:cvsread@cvs.php.net:/repository login
$ cvs -d:pserver:cvsread@cvs.php.net:/repository co pecl/extname
```

- Téléchargements pour Windows

Les utilisateurs de Windows peuvent trouver les bibliothèques compilées PECL dans la **Collection of PECL modules** disponible sur la page [téléchargements PHP](#), et en suivant le lien [PECL Snapshot](#). Pour compiler PHP sous Windows, lisez le fichier [Win32 Build README](#).

4.5.3 PECL sur Windows

Comme pour toute extension PHP DLL, l'installation est aussi simple que de copier la bibliothèque DDL de l'extension PECL dans le dossier extension_dir et de la charger dans le `php.ini`. Par exemple, ajoutez la ligne suivante dans votre `php.ini` :

```
extension=php_extname.dll
```

Après cela, relancez le serveur Web.

4.5.4 Compilations d'extensions PECL partagées avec PEAR

PEAR facilite la création d'extension PHP partagées. En utilisant la commande pear, faites ceci :

```
$ pear install extname
```

Ceci va télécharger le fichier source de l'extension **extname** , le compiler et installer le fichier `extname.so` dans votre dossier `extension_dir` . `extname.so` doit ensuite être chargé via `php.ini` .

Par défaut, la commande `pear` n'installera pas les paquets marqués comme étant alpha ou beta . Si aucun paquet stable 'est disponible, vous devriez installer un paquet beta en utilisant la commande suivante :

```
$ pear install extname-beta
```

Vous pouvez également installer une version spécifique en utilisant la commande :

```
$ pear install extname-0.1
```

4.5.5 Compilation des extensions partagées avec `phpize`

Parfois, l'utilisation de l'installateur `pear` n'est pas une bonne option. Ceci parceque vous être derrière un pare-feu ou parceque l'extension que vous voulez installer n'est pas disponible en tant que paquet PEAR comme une extension uniquement disponible via CVS. Si vous devez compiler une telle extension, vous pouvez utiliser l'utilitaire de compilation afin d'exécuter la compilation manuellement.

La commande `phpize` est utilisé pour préparer l'environnement de compilation pour une extension PHP. Dans l'exemple suivant, les sources de l'extension sont dans un dossier appelé `extname` :

```
$ cd extname
$ phpize
$ ./configure
$ make
# make install
```

Une installation réussie aura créée un fichier `extname.so` et l'aura placée dans le [dossier des extensions](#) de PHP. Vous devez ajouter la ligne `extension=extname.so` dans votre `php.ini` avant de pouvoir utiliser l'extension.

4.5.6 Compilation des extensions PECL statiquement dans PHP

Il peut arriver que vous ayez à compiler votre extension PECL statiquement dans votre binaire PHP. Pour ce faire, vous devez placer les sources de l'extension dans le dossier `php-src/ext/` et ré-exécuter le script de configuration de PHP.

```
$ cd /your/phpsrkdir/ext
$ pear download extname
$ gzip -d < extname.tgz | tar -xvf -
$ mv extname-x.x.x extname
$ rm package.xml
```

Cela générera le dossier suivant :

/your/phpsrcdir/ext/extname

A partir de la, forcez PHP à reconstruire le script de configuration, puis, suivez le processus classique de compilation de PHP :

```
$ cd /your/phpsrcdir
$ rm configure
$ ./buildconf --force
$ ./configure --help
$ ./configure --with-extname --enable-someotherext --with-foobar
$ make
$ make install
```

Note

Pour exécuter le script 'buildconf', vous devez posséder autoconf 2.13 et automake 1.4+ (les versions plus récentes de autoconf peuvent fonctionner, mais ne sont pas supportées).

L'utilisation de --enable-extname ou --with-extname dépend de l'extension. Généralement, une extension qui ne dépend pas d'une bibliothèque externe utilise --enable . Pour être certains, utilisez la commande suivante après avoir utilisé buildconf :

```
$ ./configure --help | grep extname
```

4.6 Configuration

4.6.1 Le fichier de configuration

Le fichier de configuration (appelé php3.ini dans la version 3 de PHP, et simplement php.ini dans la version 4) est lu par PHP au démarrage. Si vous avez compilé PHP en module, le fichier n'est lu qu'une seule fois, au lancement du démon HTTP . Pour les versions CGI et CLI le fichier est lu à chaque invocation.

Le php.ini est cherché dans ces endroits (et dans cet ordre) :

- L'endroit spécifique du module SAPI (la directive PHPIniDir d'Apache 2, l'option de la ligne de commande -c en CGI et en CLI, le paramètre php_ini en NSAPI, la variable d'environnement PHP_INI_PATH en THTTPD)
- HKEY_LOCAL_MACHINE\SOFTWARE\PHP\IniFilePath (Base de registre de Windows)
- La variable d'environnement PHPRC
- Le dossier courant de travail (en CLI)
- Le dossier du serveur web (pour les modules SAPI), ou le dossier contenant PHP (autrement dans Windows)
- Le dossier Windows (C:\windows ou C:\winnt) (pour Windows), ou l'option de compilation --with-config-file-path lors de la compilation

Si le fichier php-SAPI.ini existe (où SAPI utilise SAPI, donc le nom du fichier est e.g. php-cli.ini ou php-apache.ini), il sera utilisé à la place du php.ini . Le nom SAPI peut être déterminé en utilisant la fonction `php_sapi_name` .

Note

Le serveur web Apache change ce dossier en dossier root au démarrage, ce qui fait que PHP essaye de lire php.ini depuis le système de fichier racine s'il existe.

Les directives php.ini par extensions sont documentées respectivement sur les pages du manuel de ces dernières directement. La [liste des directives internes](#) est disponible en annexe. Il est probable que toutes les directives PHP ne sont pas documentées dans le manuel. Pour une liste complète des directives disponibles dans votre version de PHP, merci de lire les commentaires de votre propre fichier php.ini . Vous pouvez également trouver la [dernière liste disponible](#) sur CVS.

Extrait du php.ini

```
; tout texte sur une ligne, situé après un point-virgule ";" est ignoré
[php] ; les marqueur de section (texte entre crochets) sont aussi ignorés
; Les valeurs booléennes peuvent être spécifiées comme ceci :
;   true, on, yes
; ou false, off, no, none
register_globals = off
track_errors = yes

; vous pouvez placer les chaînes de caractères entre guillemets
include_path = "./usr/local/lib/php"

; Les anti-slash sont traités comme n'importe quel caractère
include_path = ".;c:\php\lib"
```

Depuis PHP 5.1.0, il est possible de se référer à des variables .ini depuis des fichiers .ini. Par exemple : `open_basedir = ${open_basedir} "/new/dir"` .

4.6.2 Comment modifier la configuration

4.6.2.1 Exécuter PHP comme module Apache

Lorsque vous utilisez le module Apache, vous pouvez aussi changer les paramètres de configuration en utilisant les directives dans les fichiers de configuration d'Apache (httpd.conf) et dans les fichiers .htaccess . Vous aurez besoin des privilèges "AllowOverride Options" ou "AllowOverride All".

Dans la version 4 et 5 de PHP, il y a de nombreuses directives Apache qui vous permettent de modifier la configuration de PHP à partir des fichiers de configuration Apache. Pour une liste des directives qui sont PHP_INI_ALL , PHP_INI_PERDIR ou PHP_INI_SYSTEM reportez-vous à l'appendice [Liste des directives du php.ini](#) .

Note

Dans la version 3 de PHP, à chaque directive de configuration présente dans le fichier de configuration d'Apache correspond une directive de configuration dans le fichier php3.ini , à l'exception des directives préfixées par php3_ .

php_value nom valeur

Modifie la valeur de la directive spécifiée. Cette instruction n'est utilisable qu'avec les directives PHP de type PHP_INI_ALL et PHP_INI_PERDIR . Pour annuler une valeur qui aurait été modifiée au préalable, utilisez la valeur none .

Note

N'utilisez pas php_value pour configurer des valeurs booléennes. php_flag (voir plus bas) doit être utilisée.

php_flag nom on|off

Cette instruction est utilisée pour activer ou désactiver une option. Cette instruction n'est utilisable qu'avec les directives PHP de type PHP_INI_ALL et PHP_INI_PERDIR .

php_admin_value nom valeur

Cette instruction affecte une valeur à la variable spécifiée. Cette instruction **NE peut PAS être utilisée** dans un fichier .htaccess . Toute directive de PHP configurée avec le type php_admin_value ne peut pas être modifiée en utilisant le fichier .htaccess ou les directives de virtualhost. Pour annuler une valeur qui aurait été modifiée au préalable, utilisez la valeur none .

php_admin_flag name on|off

Cette directive est utilisée pour activer ou désactiver l'option précédente. Cette instruction **NE peut PAS être utilisée** dans un fichier .htaccess . Toute directive de PHP configurée avec le type php_admin_flag ne peut pas être modifiée en utilisant le fichier .htaccess ou par une directive virtualhost.

Exemple de configuration Apache

```
<IfModule mod_php5.c>
  php_value include_path "./usr/local/lib/php"
  php_admin_flag safe_mode on
</IfModule>
<IfModule mod_php4.c>
  php_value include_path "./usr/local/lib/php"
  php_admin_flag safe_mode on
</IfModule>
<IfModule mod_php3.c>
  php3_include_path "./usr/local/lib/php"
  php3_safe_mode on
</IfModule>
```

Attention

Les constantes PHP n'existent pas en dehors de PHP. Par exemple, dans le fichier httpd.conf , vous ne pouvez pas utiliser des constantes PHP telles que E_ALL ou E_NOTICE pour fixer le niveau de rapport d'erreur , car ces constantes n'ont pas de signification pour Apache, et seront remplacées par **0** . Utilisez les valeurs numériques à la place. Les constantes peuvent être utilisées dans le php.ini

4.6.2.2 Modifier la configuration de PHP dans la base de registre Windows

Lorsque vous utilisez PHP sur Windows, la configuration peut être modifiée dossier par dossier en utilisant la base de registres de Windows. Les valeurs de configuration sont stockées avec la clé de registre HKLM\SOFTWARE\PHP\Per Directory Values , dans les sous-clés correspondantes aux noms de dossier. Par exemple, la valeur d'une option dans le dossier c:\inetpub\wwwroot sera stockée dans la clé HKLM\SOFTWARE\PHP\Per Directory Values\c:\inetpub\wwwroot . La valeur de cette option sera utilisée pour tous les scripts qui fonctionnent dans ce dossier ou ses sous-dossiers. Les valeurs sous la clé doivent avoir le nom d'une direction de configuration PHP , et la valeur correspondante. Les constantes PHP ne sont pas utilisables : il faut mettre la valeur entière. Cependant, seules les valeurs des configurations dans PHP_INI_USER peuvent être fixées de cette manière, celles dans PHP_INI_PERDIR ne peuvent l'être.

4.6.2.3 Autres interfaces de configuration de PHP

Suivant la façon dont vous exécutez PHP, vous pouvez changer certaines valeurs durant l'exécution de vos scripts en utilisant ini_set . Voir la documentation de la fonction ini_set pour plus d'informations.

Si vous êtes intéressé par une liste complète des options configurées sur votre système avec leurs valeurs courantes, vous pouvez exécuter la fonction `phpinfo` et consulter la page résultante. Vous pouvez aussi accéder individuellement aux directives de configurations pendant l'exécution de vos scripts en utilisant soit la fonction `ini_get` , soit la fonction `get_cfg_var` .

5 Référence du langage

5.1 La syntaxe de base

5.1.1 Passer du HTML au PHP

Lorsque PHP traite un fichier, il cherche les balises d'ouvertures et de fermetures, qui délimitent le code qu'il doit interpréter. De cette manière, cela permet à PHP d'être présent dans toutes sortes de documents, car tout ce qui se trouve en dehors des balises ouvrantes / fermantes de PHP est ignoré. La plupart du temps, vous verrez du code PHP dans des documents HTML, comme dans l'exemple ci-dessous.

Code PHP dans un document HTML

```
<p>Ceci sera ignoré.</p>
<?php echo 'Alors que ceci sera analysé par PHP.'; ?>
<p>Ceci sera également ignoré.</p>
```

Vous pouvez également utiliser des structures plus avancées :

Echappement avancé

```
<?php
if ($expression) {
    ?>
    <strong>Ceci est vrai.</strong>
    <?php
} else {
    ?>
    <strong>Ceci est faux.</strong>
    <?php
}
?>
```

Ceci fonctionne comme prévu parce que lorsque PHP rencontre la balise fermante `?>`, il commence simplement à afficher ce qu'il rencontre jusqu'à ce qu'il rencontre une autre balise ouvrante.

L'exemple ci-dessus est simple, bien sûr, mais pour afficher de gros blocs de textes, la mise à l'écart de ce type de bloc de l'analyseur de PHP est plus efficace que d'envoyer la totalité du texte en utilisant les fonctions `echo` ou `print`.

Il y a quatre paires différentes de balises ouvrantes / fermantes qui peuvent être utilisées dans PHP. Deux de ces balises, `<?php ?>` et `<script language="php"> </script>`, sont toujours disponibles. Les deux autres sont les balises courtes et les balises du style ASP, et peuvent être activées ou désactivées depuis le fichier de configuration `php.ini`. Cependant, malgré le fait que des personnes trouvent les balises courtes et les balises du style ASP pratiques, ils sont moins portables et donc, généralement, non recommandés.

Note

Notez également que si vous intégrez PHP dans des documents XML ou XHTML, vous devez utiliser les balises `<?php ?>` pour rester conforme aux standards.

Balises d'ouvertures et de fermetures PHP

1. `<?php echo 'Si vous voulez réaliser des documents XHTML ou XML, faites comme ceci'; ?>`
2. `<script language="php">
 echo 'quelques éditeurs (comme FrontPage)
 </script>`

```

        n\ 'aient pas ce genre d'instructions';
    </script>

```

3. `<? echo 'ceci est le plus simple, une instruction SGML'; ?>`
`<?= expression ?>` Ceci est la version courte pour "`<? echo expression ?>`"
4. `<% echo 'Vous pouvez optionnellement utiliser les balises ASP-style'; %>`
`<%= $variable; # Ceci est la version courte pour "<% echo . . ." %>`

Bien que les balises vues dans les exemples un et deux sont toutes les deux disponibles, l'exemple un est le plus communément utilisé et le plus recommandé des deux.

Les balises courtes (troisième exemple) ne sont disponibles que s'ils ont été activées via la directive `short_open_tag` du fichier de configuration `php.ini`, ou si PHP a été configuré avec l'option `--enable-short-tags`.

Note

Si vous utilisez PHP 3, vous devez également activer les balises courtes via la fonction `short_tags`. **Ceci est uniquement disponible en PHP 3 !**

Les balises du style ASP (quatrième exemple) sont uniquement disponibles lorsqu'elles sont activées via la directive `asp_tags` du fichier de configuration `php.ini`.

Note

Le support des balises style ASP a été ajouté en PHP 3.0.4.

Note

L'utilisation des balises courtes doit être banni lors de développements d'applications ou de bibliothèques qui sont destinées à être redistribuées, ou déployées sur des serveurs qui ne sont pas sous votre contrôle, car les balises courtes peuvent ne pas être supportées sur le serveur cible. Pour réaliser du code portable, redistribuable, n'utilisez jamais les balises courtes.

5.1.2 Séparation des instructions

Comme en C ou en Perl, PHP requiert que les instructions soient terminées par un point-virgule à la fin de chaque instruction. La balise fermante d'un bloc de code PHP implique automatiquement un point-virgule ; vous n'avez pas besoin d'utiliser un point-virgule pour terminer la dernière ligne d'un bloc PHP. La balise fermante d'un bloc inclura immédiatement un caractère de nouvelle ligne si un est présent.

Séparation des instructions

```

<?php
    echo 'Ceci est un test';
?>

<?php echo 'Ceci est un test' ?>

<?php echo 'Oubli de la balise fermante';

```

Note

La balise fermante d'un bloc PHP à la fin d'un fichier est optionnel, et parfois, il est utile de l'omettre lors de l'utilisation de la fonction `include` ou de la fonction `require`, car les espaces non désirés n'apparaîtront pas à la fin des fichiers, et ainsi, vous pourrez toujours ajouter des en-têtes à la réponse plus tard. C'est utile également si vous voulez utiliser l'affichage du buffer et que vous ne voulez pas voir d'espaces blancs ajoutés à la fin des parties générées par les fichiers inclus.

5.1.3 Commentaires

PHP supporte les commentaires 'C', 'C++' et Unix shell-style (style Perl). Par exemple :

Exemple de commentaire

```
<?php
    echo 'Ceci est un test'; // Ceci est un commentaire sur une seule ligne, style c++
    /* Ceci est un commentaire sur
       plusieurs lignes */
    echo 'Ceci est un autre test';
    echo 'Et un test final'; # Ceci est un commentaire style shell sur une seule ligne
?>
```

Les commentaires sur une seule ligne comment jusqu'à la fin de la ligne ou le bloc courant de code PHP, le premier des deux. Ceci signifie que le code HTML après // ... ?> ou après # ... ?> SERA affiché : ?> terminera le mode PHP et retournera en mode HTML, et // ou # n'influencera pas cela. Si la directive de configuration `asp_tags` est activée, ce comportement sera identique avec // %> et # %> . Cependant, la balise `</script>` ne terminera pas le mode PHP dans un commentaire d'une seule ligne.

Les commentaires vont jusqu'à la fin de la ligne

```
<h1>Ceci est un exemple <?php # echo 'simple';?>.</h1>
<p>La ligne ci-dessus affichera 'Ceci est un exemple'.</p>
```

Les commentaires style 'C' commentent jusqu'à ce que le premier */ soit rencontré. Vous devriez faire attention aux commentaires style 'C' nichés dans de gros blocs lorsque vous les commentez.

Les commentaires de type C

```
<?php
/*
    echo 'Ceci est un test'; /* Ce commentaire posera un problème */
*/
?>
```

5.2 Les types

5.2.1 Introduction

PHP supporte les huit types basiques suivants :

PHP supporte quatre types scalaires :

- booléen
- entier
- nombre à virgule flottante
- chaîne de caractères

PHP supporte deux types composés :

- tableau
- objet

PHP supporte deux types spéciaux :

- [ressource](#)
- NULL

Ce manuel introduit également quelques [pseudo-types](#) pour des raisons de lisibilités :

- mixed
- nombre
- callback

Vous pouvez également trouver quelques références au type "double". Considérez le type "double" comme le type "float" (nombres décimaux), les deux noms existent uniquement pour des raisons historiques.

Habituellement, le type d'une variable n'est pas déclaré par le programmeur. Il est décidé au moment de l'exécution par PHP, en fonction du contexte dans lequel la variable est utilisée.

<p>Note</p> <p>Si vous voulez vérifier le type et la valeur d'une variable ou d'une expression , utilisez la fonction var_dump .</p> <p>Si vous souhaitez simplement une représentation lisible pour du débogage, utilisez la fonction gettype . Pour vérifier la présence de certains types, n'utilisez pas gettype , mais plutôt les fonctions is_ type . Par exemple :</p>
<p>Exemples de vérification de type de variable</p> <pre><?php \$bool = TRUE; // un booléen \$str = 'foo'; // une chaîne de caractères \$int = 12; // un entier echo gettype(\$bool); // affiche "boolean" echo gettype(\$str); // affiche "string" // Ceci est un entier, on l'incrmente de 4 if (is_int(\$int)) { \$int += 4; } // Si \$bool est une chaîne, l'afficher // (Ne pas imprimer n'importe quoi). if (is_string(\$bool)) { echo 'Chaîne : ' . \$bool; } ?></pre>

Si vous voulez forcer une variable à être convertie en un certain type, vous devez transtyper ([cast](#)) la variable ou utiliser la fonction [settype](#) .

Il est à noter qu'une variable peut se comporter de manière différente suivant les situations, en fonction du type qui lui est affectée. Pour plus d'informations, voir le paragraphe sur le [transtypage](#) . Vous pouvez également être intéressé par la [table de comparaison des types](#) , qui vous montrent des exemples variés concernant les relations de comparaisons des types.

5.2.2 Booléens

C'est le type le plus simple. Un booléen exprime une valeur de vérité. Il peut prendre comme valeur soit TRUE soit FALSE .

Note

Le type booléen a été introduit en PHP 4.

5.2.2.1 Syntaxe

Vous pouvez utiliser les mots-clés ' TRUE ' et ' FALSE ' pour spécifier une valeur de type booléen . Ces mots-clés sont insensibles à la casse.

Utilisation des mots-clés TRUE et FALSE

```
<?php
$foo = True; // assigne la valeur TRUE à la variable $foo
?>
```

Généralement, vous les créez avec un des opérateurs qui retourne une valeur bool , pour les passer à une structure de contrôle .

Utilisation des constantes TRUE et FALSE avec une structure de contrôle

```
<?php
// == est un opérateur d'égalité
// qui retourne un booléen
if ($action == 'show_version') {
    echo 'La version est la 1.23';
}

// Ceci n'est pas nécessaire
if ($show_separators == TRUE) {
    echo "<hr />\n";
}

// car vous pouvez simplement utiliser
if ($show_separators) {
    echo "<hr />\n";
}
?>
```

5.2.2.2 Conversion en booléen

Pour explicitement convertir une valeur en booléen , utilisez les opérateurs de transtypage (bool) ou (boolean) . Cependant, dans la plupart des cas, vous n'avez pas besoin d'utiliser ces opérateurs de transtypage, car une valeur sera automatiquement convertie si un opérateur, une fonction ou une structure de contrôle requièrent un argument de type booléen .

Voir aussi le chapitre Définition du type .

Lors des conversions de valeurs de type bool , les valeurs suivantes sont considérées comme fausses (FALSE) :

- Le booléen FALSE lui-même
- L' entier 0 (zéro)
- La chaîne de caractères 0.0 (zéro)
- La chaîne de caractères vide et la chaîne de caractères "0"

- Le tableau vide (aucun élément)
- L' objet vide (aucun élément) (PHP 4 uniquement)
- Le type spécial NULL

Toutes les autres valeurs sont considérées comme vraies (TRUE) (y compris les ressources).

Attention
-1 est considéré comme vrai, tout comme les nombres autres que zéro (aussi bien positifs que négatifs) !
Conversion en <u>booléen</u>
<pre><?php var_dump((bool) ""); // bool(false) var_dump((bool) 1); // bool(true) var_dump((bool) -2); // bool(true) var_dump((bool) "foo"); // bool(true) var_dump((bool) 2.3e5); // bool(true) var_dump((bool) array(12)); // bool(true) var_dump((bool) array()); // bool(false) var_dump((bool) "false"); // bool(true) ?></pre>

5.2.3 Entiers

Un entier est un nombre de l'ensemble des entiers naturels $Z : Z = \{ \dots, -2, -1, 0, 1, 2, \dots \}$.

Voir aussi : Longueur arbitraire des entiers / GMP , les nombre à virgule flottante et les précisions arbitraires / BCMath .

5.2.3.1 Syntaxe

Les entiers peuvent être spécifiés en base décimale (dite aussi base 10), en hexadécimale (base 16) ou octale (base 8). Les entiers peuvent être optionnellement précédés par le signe plus ou moins (- ou +).

Pour utiliser la notation octale, vous devez préfixer le nombre avec un zéro; pour utiliser la notation hexadécimale, vous devez préfixer le nombre avec 0x .

Notations entières
<pre><?php \$a = 1234; // nombre entier en base 10 \$a = -123; // nombre entier négatif \$a = 0123; // nombre entier en base 8, octale (équivalent à 83 en base 10) \$a = 0x1A; // nombre entier en base 16, hexadécimale // (équivalent à 26 en base 10) ?></pre>

Techniquement, la structure d'un entier suit les règles suivantes :

Structure d'un entier
<pre>decimal : [1-9][0-9]* 0 hexadecimal : 0[xX][0-9a-fA-F]+ octal : 0[0-7]+ integer : [+]?decimal</pre>

```
| [+]?hexadecimal
| [+]?octal
```

La taille des entiers dépend de la plate-forme de support, mais la valeur maximale est généralement de 2 milliards et des poussières (c'est un entier signé de 32 bits). PHP ne supporte pas les entiers non signés.

Attention

Si un caractère invalide est utilisé dans un entier octal (par exemple, 8 ou 9), le reste du nombre est ignoré.

Interprétation des nombres octaux

```
<?php
var_dump(01090); // 010 octal = 8 decimal
?>
```

5.2.3.2 Dépassement de capacité des entiers

Si un nombre est hors de l'intervalle de validité des entiers, il sera interprété comme un float . De même, si vous effectuez une opération qui retourne un nombre hors de l'intervalle de validité des entiers, un float sera retourné.

Dépassement de capacité des entiers

```
<?php
$large_number = 2147483647;
var_dump($large_number);
// affiche : int(2147483647)

$large_number = 2147483648;
var_dump($large_number);
// affiche : float(2147483648)

// cette méthode ne convient pas pour les entiers hexadécimaux :
var_dump(
Notice: Hex number is too big: 0x100000000 in /Volumes/ibook/phpdoc-fr/pdf.inc on line 627
0x100000000 );
// affiche : int(2147483647)

$million = 1000000;
$large_number = 50000 * $million;
var_dump($large_number);
// affiche : float(50000000000)
?>
```

Attention

Malheureusement, il y a un bogue dans le moteur (corrigé en 4.1.0), qui fait que PHP ne fonctionne pas toujours bien lorsque des nombres négatifs sont utilisés. Lorsque les deux opérandes sont positifs, il n'y a pas de problèmes. Par exemple : `-50000 * $million` , conduit à `-429496728` . Cependant, lorsque les deux opérateurs sont positifs, il n'y a aucun problème.

Ceci a été résolu en PHP 4.1.0.

Il n'existe pas de division entière en PHP. `1/2` générera le float `0.5` . Vous pouvez forcer la valeur au format entier pour faire l'arrondi ou bien utiliser la fonction `round` .

Divisions entières

```
<?php
var_dump(25/7); // float(3.5714285714286)
var_dump((int) (25/7)); // int(3)
var_dump(round(25/7)); // float(4)
```

```
?>
```

5.2.3.3 Conversion en entiers

Pour explicitement convertir une valeur en entier , utilisez les opérateurs de transtypage (int) ou (integer) . Cependant, dans la plupart des situations, vous n'en aurez pas besoin, car une valeur sera automatiquement convertie si un opérateur, une fonction ou tout autre élément du langage requiert un entier . Vous pouvez également convertir une valeur en entier avec la fonction intval .

Voir aussi la section sur les définitions de types .

5.2.3.3.1 Depuis un booléen

FALSE devient 0 (zéro), et TRUE devient 1 (un).

5.2.3.3.2 Depuis un nombre à virgule flottante

Lors de conversion entre un nombre décimal et un entier, le nombre sera arrondi à la valeur inférieure s'il est positif, et supérieure s'il est négatif (conversion dite 'vers zéro').

Si le nombre est hors de l'intervalle de validité des entiers, (généralement +/- 2.15e+9 = 2³¹), le résultat est indéfini, car les nombres à virgule flottante n'ont pas assez de précision pour fournir une valeur exacte pour un entier. Aucune alerte, même pas le plus petit message ne sera affiché dans ce cas !

Attention

Ne transformez jamais une fraction inconnue en entier, car cela peut conduire à des résultats irrationnels.

Attention aux erreurs de fractions

```
<?php
    echo (int) ( (0.1+0.7) * 10 );
// affiche 7!
?>
```

Pour plus d'informations, reportez-vous aux alertes liées aux nombres décimaux.

5.2.3.3.3 A partir de chaînes de caractères

Reportez-vous à la section des conversions de chaînes .

5.2.3.3.4 Conversion d'autres types

La conversion d'autres types en entier est indéfinie. En fait, PHP convertit d'abord la valeur en booléen .

Attention

Le comportement de la conversion en un entier est indéfini pour les autres types. En fait, le comportement est le même que si la valeur a d'abord été convertie en booléen . Cependant, **ne vous fiez pas** à ce comportement car il peut changer à tout moment.

5.2.4 Les nombres décimaux

Les nombres décimaux (connus aussi sous le vocable de " double ", " float " ou " nombre réels ") peuvent être spécifiés en utilisant la syntaxe suivante :

Exemple de nombres décimaux

```
<?php
$a = 1.234;
$b = 1.2e3;
$c = 7E-10;
?>
```

Formellement :

```
LNUM      [0-9]+
DNUM      ([0-9]*[\.]{LNUM}) | ({LNUM}[\.][0-9]*)
EXPONENT_DNUM ( ({LNUM} | {DNUM}) [eE][+-]? {LNUM})
```

La taille des nombres décimaux dépend de la plate-forme. Un maximum de $\sim 1.8e308$ avec une précision de 14 décimales est une configuration répandue : c'est le format 64 bits IEEE.

Attention

Précision des nombres décimaux

Il est fréquent que de simples fractions décimales telles que 0.1 ou 0.7 ne puissent être converties au format interne binaire sans une légère perte de précision. Cela peut conduire à des résultats étonnants : par exemple, `floor((0.1+0.7)*10)` retournera 7 au lieu de 8 car le résultat de la représentation interne est 7.9999999999...

Tout ceci est lié au fait qu'il est impossible d'exprimer certaines fractions en un nombre fini de chiffres. Par exemple 1/3 s'écrira 0.3333333... en mode décimal.

Ne faites donc jamais confiance aux nombres à virgule jusqu'à leur dernière décimale et ne comparez jamais ces nombres avec l'opérateur d'égalité. Si vous avez besoin d'une précision particulière, reportez-vous au traitement des nombres de grande taille avec les bibliothèques [BC](#) ou [GMP](#).

5.2.4.2 Conversion d'un nombre décimal

Pour savoir quand et comment des chaînes sont converties en nombres à virgules flottantes, voyez la section "[Conversion de chaînes de caractères](#)". Pour les valeurs d'autres types, la conversion est la même que si la valeur était d'abord convertie en entier, puis en nombre décimal. Voyez la section "[Conversion d'un entier](#)". Depuis PHP 5, une alerte est affichée si vous convertissez un objet en nombre décimal.

5.2.5 Les chaînes de caractères

Les chaînes de caractères sont des séquences de caractères. En PHP, un caractère est un octet et il y en a 256 de possibles. PHP n'a pas (encore?) de support natif d'Unicode. Voir les fonctions [utf8_encode](#) et [utf8_decode](#) pour un support Unicode.

Note

La taille n'est pas un problème majeur pour une chaîne. Elle peut devenir très grande sans problème. Il n'y a pas à s'en faire pour cela.

5.2.5.1 Syntaxe

Une chaîne peut être spécifiée de trois manières différentes :

- guillemets simples
- guillemets doubles
- syntaxe HereDoc

5.2.5.1.1 Guillemets simples

Le moyen le plus simple de spécifier une chaîne de caractères est d'utiliser les guillemets simples : '

Pour spécifier un guillemet simple littéral, vous devez l'échapper avec un anti-slash (\), comme dans de nombreux langages. Si un anti-slash doit apparaître dans votre chaîne ou bien en fin de chaîne, il faudra le doubler. Notez que si vous essayez d'échapper n'importe quel autre caractère, l'anti-slash sera conservé! Il n'y a pas besoin d'échapper d'autres caractères que le guillemet lui-même.

<p>Note</p> <p>En PHP 3, une alerte sera affichée si cela arrive avec un niveau de rapport d'erreur de E_NOTICE</p> <p>.</p>
<p>Note</p> <p>Contrairement aux autres syntaxes, les <u>variables</u> présentes dans la chaîne ne seront PAS remplacées par leurs valeurs.</p>
<p>Exemples d'interpolation des chaînes de caractères</p> <pre style="font-family: monospace; font-size: 0.9em; color: #008000;"> <?php echo 'Ceci est une chaîne simple'; echo 'Vous pouvez inclure des nouvelles lignes dans une chaîne, comme ceci.'; echo 'Arnold a coutume de dire : "I\'ll be back"'; // affiche : "I'll be back" echo 'Êtes-vous sûr de vouloir effacer le dossier C:*..*?'; // affiche : Êtes-vous sûr de vouloir effacer le dossier C:*..*? echo 'Êtes-vous sûr de vouloir effacer le dossier C:*..*?'; // affiche : Êtes-vous sûr de vouloir effacer le dossier C:*..*? echo 'Je suis en train de mettre une nouvelle ligne comme ceci : \n'; // affiche : Je suis en train de mettre une nouvelle ligne comme ceci : \n echo 'Les variables ne seront pas \$afficher \$ici'; // affiche : Les variable ne seront pas \$afficher \$ici ?></pre>

5.2.5.1.2 Guillemets doubles

Si la chaîne est entourée de guillemets doubles ("), PHP va comprendre certaines séquences de caractères :

Séquence	Valeur
\n	Nouvelle ligne (linefeed , LF ou 0x0A (10) en ASCII)
\r	Retour à la ligne (carriage return , CR ou 0x0D (13) en ASCII)
\t	Tabulation horizontale (HT ou 0x09 (9) en ASCII)
\\	Anti-slash
\\$	Caractère \$
\"	Guillemets doubles
\[0-7]{1,3}	Une séquence de caractères qui permet de rechercher un nombre en notation octale.
\x[0-9A-Fa-f]{1,2}	Une séquence de caractères qui permet de rechercher un nombre en notation hexadécimale.

Si vous essayez d'utiliser l'anti-slash sur n'importe quelle autre séquence, l'anti-slash sera affiché dans votre chaîne.

Le plus important pour les chaînes à guillemets doubles est le fait que les variables qui s'y trouvent seront remplacées par leur valeur. Voir la section sur le [traitement des variables dans les chaînes](#) pour plus de détails.

5.2.5.1.3 Syntaxe Heredoc

Un autre moyen de délimiter les chaînes est d'utiliser la syntaxe dite " Heredoc " (en français, documentation [ici](#)): <<<, suivi d'un identifiant arbitraire, puis de la chaîne, puis l'identifiant initial pour terminer la séquence.

L'identifiant fermant **doit** commencer à la première colonne de la ligne. L'identifiant utilisé doit suivre les mêmes règles que les étiquettes PHP : il ne doit contenir uniquement que des caractères alpha-numériques, et des soulignés ("_"), et enfin, commencer par un caractère alphabétique ou un souligné.

Attention
<p>Il est très important de noter que la ligne qui contient l'identifiant de fermeture ne doit contenir aucun autre caractère, hormis, éventuellement, un point-virgule (;). Cela signifie notamment que l'identifiant ne doit pas être indenté et qu'il n'y a aucun caractère d'espacement ou de tabulation avant ou après le point virgule. Il est également important de noter que le premier caractères avant l'identifiant de fermeture doit être une nouvelle ligne, tel que définie par votre système. Par exemple, ce sera \r sous Macintosh.</p> <p>Si cette règle n'est pas respectée et que l'identifiant de fermeture n'est pas correct, alors, il ne sera pas considéré comme un identifiant fermant et PHP continuera d'en chercher un. Si dans ce cas aucun identifiant de fermeture n'est trouvé, une erreur d'analyse sera émise, contenant comme numéro de ligne la fin du script.</p> <p>Il n'est pas autorisé d'utiliser la syntaxe " Heredoc " lors de l'initialisation des membres d'une classe. Utilisez une autre syntaxe pour les chaînes à la place.</p>
Exemple incorrect

```
<?php
class foo {
    public $bar = <<<EOT
bar
EOT;
}
?>
```

La syntaxe Heredoc se comporte exactement comme une chaîne à guillemets doubles, sans les guillemets doubles. Cela signifie que vous n'avez pas à échapper les guillemets (simples ou doubles) dans cette syntaxe. Les variables sont remplacées par leur valeur et le même soin doit leur être apporté que dans les chaînes à guillemets doubles.

Exemple de chaîne HereDoc

```
<?php
$str = <<<EOD
    Exemple de chaîne
    s'étalant sur
    plusieurs lignes
    avec la syntaxe heredoc
EOD;

/* Exemple plus complexe, avec des variables. */
class foo {
    var $foo;
    var $bar;
    function foo() {
        $this->foo = 'Foo';
        $this->bar = array('Bar1', 'Bar2', 'Bar3');
    }
}

$foo = new foo();
$name = 'MonNom';

echo <<<EOT
    Mon nom est "$name". J'affiche des $foo->foo.
    Maintenant, j'affiche un {$foo->bar[1]}.
    Ceci se traduit par un 'A' majuscule : \x41
EOT;
?>
```

Note

Le support Here doc a été ajouté en PHP 4.

5.2.5.1.4 Traitement des variables dans les chaînes

Lorsqu'une chaîne est spécifiée avec des guillemets doubles ou en utilisant la syntaxe heredoc, les variables qu'elle contient sont remplacées par leur valeur.

Il y a deux types de syntaxe, une simple et une complexe. La syntaxe simple est la plus courante et la plus pratique : elle fournit un moyen d'utiliser les variables, que ce soient des chaînes, des tableaux ou des membres d'objets.

La syntaxe complexe a été introduite en PHP 4 et peut être reconnue grâce aux accolades entourant les expressions.

5.2.5.1.4.1 Syntaxe simple

Dès qu'un signe dollar \$ est rencontré, l'analyseur PHP va lire autant de caractères qu'il peut pour former un nom de variable valide. Entourez le nom de la variable avec des accolades pour indiquer explicitement son nom.

Utilisation des accolades {} dans les chaînes

```
<?php
$boisson = 'vin';
echo "Du $boisson, du pain et du fromage!";
// Correct, car "," n'est pas autorisé dans les noms de variables

echo 'Il a goûté plusieurs ' . $boissons;
// Pas correct, car 's' peut faire partie d'un nom de variable, et PHP recherchera alors $boiss

echo "Il a goûté plusieurs ${boisson}s";
// Correct
?>
```

De la même façon, vous pouvez utiliser un élément de tableau ou un membre d'objet. Pour les éléments de tableau, le crochet fermant ']' marquera la fin du nom de la variable. Pour les membres d'objets, les mêmes règles que ci-dessus s'appliquent. Cependant, il n'existe pas d'astuce comme cela pour les variables simples.

Utilisation des accolades {} dans les chaînes pour les tableaux

```
<?php
// Ces exemples sont spécifiques à l'utilisation de tableaux dans une chaîne.
// Lorsque vous êtes hors d'une chaîne, utilisez toujours des guillemets
// autour des index de tableau, et n'utilisez pas d'accolades}.

// Affichons toutes les erreurs
error_reporting(E_ALL);

$fruits = array('fraise' => 'rouge', 'banane' => 'jaune');

// Fonctionne mais notez que cela fonctionne pas comme
// si cela était hors d'une chaîne
echo "Une banane est $fruits[banane].";

// Fonctionne
echo "Une banane est {$fruits['banane']}.";

// Fonctionne mais PHP cherche une constante appelée banane
// tel que décrit ci-dessous
echo "Une banane est {$fruits[banane]}.";

// Ne fonctionne pas, il manque les accolades. Cela donne une erreur d'analyse
echo "Une banane est $fruits['banane'].";

// Fonctionne
echo "Une banane est " . $fruits['banane'] . ".";

// Fonctionne
echo "Ce carré a un coté de $$square->width mètres de large.";

// Ne fonctionne pas. Pour une solution, voyez la syntaxe complexe.
echo "Ce carré a un coté de $$square->width00 centimètres.";
?>
```

Pour tout ce qui sera plus compliqué, voyez la syntaxe complexe.

5.2.5.1.4.2 Syntaxe complexe

La syntaxe est dite "complexe" car elle permet l'utilisation d'expressions complexes et non pas parce qu'elle serait obscure.

En fait, vous pouvez inclure n'importe quelle valeur qui est dans l'espace de nom avec cette syntaxe. Il suffit d'écrire une expression exactement comme si elle était hors de la chaîne, puis de l'entourer d'accolades {}. Puisque vous ne pouvez pas échapper les accolades, cette syntaxe ne commence qu'à partir du signe dollar, qui suit immédiatement l'accolade ouvrante. Vous pouvez utiliser "\\$" ou "\{" pour obtenir un "{" littéral. Voici quelques exemples :

Syntaxe complexe de chaîne
<pre> <?php // Affichons toutes les erreurs error_reporting(E_ALL); \$super = 'fantastique'; // Ne fonctionne pas. Affiche : Ceci est { fantastique} echo "Ceci est { \$super}"; // Fonctionne. Affiche Ceci est fantastique echo "This is {\$super}"; echo "This is \${super}"; // Fonctionne echo "Ce carré a un coté de {\$square->width}00 centimètres."; // Fonctionne echo "Ceci fonctionne : {\$arr[4][3]}"; // Ceci est faux pour la même raison que \$foo[bar] est faux // hors d'une chaîne. En d'autres termes, cela va fonctionner // car PHP recherche d'abord une constante appelée foo, mais // il générera une note E_NOTICE (undefined constant). echo "Ceci est faux : {\$arr[foo][3]}"; // Fonctionne. Lorsque vous utilisez un tableau multidimensionnel dans // une chaîne, n'oubliez jamais les accolades. echo "Ceci fonctionne : {\$arr['foo'][3]}"; // Fonctionne echo "Ceci fonctionne : " . \$arr['foo'][3]; echo "Vous pouvez même écrire {\$obj->values[3]->name}"; echo "Ceci est une valeur de variable variable : {\${\$name}}"; ?> </pre>

5.2.5.1.5 Accès et modification des caractères d'une chaîne

Les caractères d'une chaîne sont accessibles et modifiables en spécifiant leur offset (le premier caractère est d'offset 0) entre accolade, après le nom de la variable.

Note
<p>Pour assurer la compatibilité ascendante, vous pouvez toujours accéder aux caractères avec des crochets. Mais cette syntaxe est obsolète en PHP 4.</p>

Exemples de chaînes

```

<?php
// Lit le premier caractère de la chaîne
$str = 'Ceci est un test.';
$first = $str{0};

// Lit le troisième caractère de la chaîne
$third = $str{2};

// Lit le dernier caractère de la chaîne
$str = 'Ceci est un test.';
$last = $str{strlen($str)-1};

// Modifie le dernier caractère de la chaîne
$str = 'Ceci est un test.';
$str{strlen($str)-1} = '!';

?>

```

5.2.5.2 Fonctions et opérateurs pratiques

Les chaînes peuvent être concaténées avec l'opérateur '.' (point). Notez que l'opérateur d'addition '+' (plus) ne fonctionnera pas. Reportez-vous à la section [opérateurs de chaînes](#) pour plus de détails.

Il y a une grande quantité de fonctions pratiques pour modifier les chaînes.

Reportez-vous à la section [chaînes de caractères](#) pour les fonctions les plus générales, à [Expressions rationnelles Perl](#) et [Expressions rationnelles POSIX étendues](#) pour les recherches et remplacements.

Il y a aussi les fonctions sur les [URL](#), ainsi que des fonctions de chiffrement ([mcrypt](#) et [mhash](#)).

Finalement, si vous ne trouvez toujours pas votre bonheur, il y a les fonctions de [types de caractères](#).

5.2.5.3 Conversion en une chaîne de caractères

Vous pouvez convertir une valeur en une [chaîne de caractères](#) en utilisant l'opérateur (string) ou bien la fonction [strval](#). La conversion en [chaîne de caractères](#) est automatiquement faite dans toutes les situations qui nécessitent une [chaîne de caractères](#). Cela arrive lorsque vous utilisez des fonctions comme [echo](#) ou [print](#) ou encore lorsque vous comparez une valeur avec une [chaîne de caractères](#). Lisez les sections du manuel sur les [types](#) et la [définition du type](#) pour comprendre plus facilement ce qui va suivre. Voir aussi la fonction [settype](#).

Le [booléen](#) TRUE est converti en la [chaîne de caractères](#) "1", et le [booléen](#) FALSE est représenté par la [chaîne de caractères](#) vide "". De cette façon, vous pouvez convertir des booléens en [chaîne de caractères](#) et vice-versa.

Un [entier](#) ou un [nombre à virgule flottante](#) est converti en une [chaîne de caractères](#) qui représente ce nombre en décimal (y compris l'exposant des nombres décimaux).

Les tableaux sont toujours convertis en la [chaîne de caractères](#) "Array", ce qui fait que vous ne pouvez pas afficher le contenu d'un tableau avec la fonction [echo](#) ou [print](#). Pour voir un seul élément, vous devriez faire quelque chose comme : `echo $arr['foo']`. Voyez les informations ci-dessous pour plus de conseils.

Les objets sont automatiquement convertis en la chaîne de caractères "Object" . Si vous souhaitez afficher le membre d'un objet, lisez le paragraphe ci-dessous. Si vous voulez connaître le nom de la classe de cet objet, utilisez la fonction `get_class` . Depuis PHP 5, la méthode `__toString()` est utilisée si elle existe.

Les ressources sont toujours converties en chaîne de caractères "Resource id #1" où 1 est le nombre unique représentant la ressource , assigné par PHP au moment de l'exécution. Si vous souhaitez connaître le type d'une ressource, utilisez get_resource_type .

NULL est toujours converti en une chaîne de caractères vide.

Comme vous pouvez le voir, afficher des tableaux, des objets ou des ressources n'est pas pertinent par cette méthode. Reportez-vous aux fonctions `print_r` et `var_dump` pour avoir des outils plus pratiques.

Vous pouvez aussi convertir des valeurs PHP en chaînes pour les sauver dans un stockage. Cette méthode est appelée la linéarisation et peut être exploitée avec la fonction `serialize` . Vous pouvez aussi linéariser des valeurs PHP en structure XML si vous avez le support WDDX dans votre configuration de PHP.

5.2.5.4 Conversion de chaînes de caractères

Lorsqu'une chaîne de caractères est évaluée comme une valeur numérique, le résultat et le type de la variable sont déterminés comme suit.

La chaîne de caractères est de type "double" si elle contient un des caractères '.', 'e' ou 'E'. Sinon, elle est de type entier ("integer").

La valeur est définie par la première partie de la chaîne. Si la chaîne de caractères débute par une valeur numérique cette valeur sera celle utilisée. Sinon, la valeur sera égale à 0 (zéro). Les données numériques valides ont un signe optionnel, suivies par un ou plusieurs digits (optionnellement, contenant un point de décimal), suivies par un exponentiel optionnel. L'exponentiel peut être un 'e' ou un 'E' suivi par un ou plusieurs digits.

Exemples de conversions automatiques

```
<?php
$foo = 1 + "10.5";           // $foo est du type float (11.5)
$foo = 1 + "-1.3e3";        // $foo est du type float (-1299)
$foo = 1 + "bob-1.3e3";    // $foo est du type integer (1)
$foo = 1 + "bob3";         // $foo est du type integer (1)
$foo = 1 + "10 petits cochons"; // $foo est du type integer (11)
$foo = 4 + "10.2 gros cochons"; // $foo est du type float (14.2)
$foo = "10.0 cochons " + 1; // $foo est du type float (11)
$foo = "10.0 cochons " + 1.0; // $foo est du type float (11)
?>
```

Pour plus d'informations sur les conversions de type, voir les pages de man Unix à propos de la fonction `strtod(3)` .

Si vous voulez tester l'un des exemples de cette section, vous pouvez faire un copier/coller et l'insérer dans un script pour voir comment il se comporte :

Exemples de conversions

```
<?php
echo "\$foo==\$foo; type is " . gettype( $foo ) . "<br />\n";
```

```
?>
```

Ne vous attendez pas à récupérer le code d'un caractère en le convertissant en un entier (comme vous pourriez le faire en C par exemple). Utilisez les fonctions `ord` et `chr` pour convertir les caractères en leurs codes.

5.2.6 Les tableaux

Un tableau PHP est en fait une association ordonnée (littéralement, une map). Une association est un type qui fait correspondre des valeurs à des **clés**. Ce type est optimisé de diverses façons, qui font que vous pouvez le manipuler comme un tableau à indices réels, une liste (vecteur), ou un table de hachage (qui est une implémentation d'association), dictionnaire, collection, pile, queue et encore d'autres. Comme une valeur peut elle-même être un tableau, vous pouvez simuler facilement un arbre.

Les détails d'implémentation de ces structures sont hors du champ de ce manuel, mais vous trouverez ici un exemple de toutes ces structures.

5.2.6.1 Syntaxe

5.2.6.1.1 Créer un tableau array

Un tableau array peut être créé avec la fonction `array`. Cette fonction prend en argument des structures `key => value`, séparées par des virgules.

`array(Array value , ...)` // `key` est soit une chaîne de caractères soit un entier positif // `value` peut être n'importe quoi

Affectations de tableaux

```
<?php
$arr = array("foo" => "bar", 12 => true);

echo $arr["foo"]; // bar
echo $arr[12];    // 1
?>
```

Une clé `key` est soit un entier soit une chaîne de caractères. Si une clé est la représentation standard d'un entier positif, elle sera interprétée comme tel. (i.e. '8' sera interprété comme 8, tandis que '08' sera interprété comme '08'). Les nombres à virgule flottante dans `key` seront transtypés en entier. Il n'y a aucune différence entre les tableaux indexés et les tableaux associatifs en PHP ; il n'y a qu'un type `array`, qui peut contenir soit des entiers, soit des chaînes en guise d'index.

Une valeur de tableau peut être de n'importe quel type.

Création d'un tableau associatif

```
<?php
$arr = array("untableau" => array(6 => 5, 13 => 9, "a" => 42));

echo $arr["untableau"][6]; // 5
echo $arr["untableau"][13]; // 9
echo $arr["untableau"]["a"]; // 42
?>
```

Si vous omettez une clé lors de la spécification d'un tableau, l'indice maximum + 1 sera utilisé comme clé par défaut. Si aucun indice numérique n'a été généré, ce sera 0. Si vous en spécifiez une qui a déjà été assignée, la nouvelle valeur écrasera la précédente.

Indexation automatique de tableau

```
<?php
// Ce tableau est identique à
array(5 => 43, 32, 56, "b" => 12);

// Celui ci
array(5 => 43, 6 => 32, 7 => 56, "b" => 12);
?>
```

Attention

Depuis PHP 4.3.0, le comportement de la génération des index décrit plus haut a changé. Maintenant, si vous ajoutez une rangée à un tableau dans lequel la clé maximale est négative, alors, la prochaine clé créée sera zéro (0). Avant, le nouvel index était défini à l'index maximal plus un, le même comportement que lorsque l'indice est positif.

Utiliser TRUE comme clé revient à utiliser le type entier de valeur 1 . Utiliser FALSE comme clé revient à utiliser le type entier de valeur 0 . Utiliser NULL comme clé revient à utiliser le type chaîne de caractères de valeur "" . Utiliser la chaîne vide comme index va créer cet index, et ce n'est pas la même chose que d'utiliser des crochets vides.

Il n'est pas possible d'utiliser des tableaux ou des objets comme clés. Ce faisant, vous génèreriez une alerte : Illegal offset type .

5.2.6.1.2 La syntaxe à crochets

Vous pouvez aussi modifier un tableau existant en lui assignant simplement des valeurs.

L'assignation de valeurs de tableau se fait en spécifiant la clé entre crochets. Si vous omettez la clé (" \$tableau[] "), la valeur sera ajoutée à la fin du tableau. \$arr[key] = valeur ; \$arr[] = valeur ; // key est soit une chaîne de caractères, soit un entier // valeur peut être n'importe quoi Si \$arr n'existe pas, il sera créé. Cela en fait une alternative pour créer un tableau. Pour modifier une valeur, assignez-lui une nouvelle valeur. Pour supprimer une valeur, utilisez la fonction unset .

Manipuler des éléments de tableau

```
<?php
$arr = array(5 => 1, 12 => 2);

$arr[] = 56; // Ceci revient à $arr[13] = 56;
           // à ce moment du script

$arr["x"] = 42; // Ceci ajoute un nouvel élément
              // avec l'index "x"

unset($arr[5]); // Ceci efface un élément du tableau

unset($arr); // Ceci efface tout le tableau
?>
```

Note

Comme indiqué ci-dessus, si vous fournissez des crochets sans spécifier de clé, le plus grand index entier existant est recherché et la nouvelle clé générée est cette valeur maximum + 1. Si aucune clé entière n'existe, la valeur sera alors 0 . Si vous spécifiez une clé qui est déjà existante, vous écraserez le contenu précédent.

Attention

Depuis PHP 4.3.0, la génération d'index a changé. Si vous ajoutez un élément et que la valeur maximale de la clé est négative, alors la prochaine clé générée est 0 . Auparavant, c'était la plus grande valeur + 1, négative ou pas. Cela reste le comportement pour les clés positives.

Notez que le maximum théorique **n'a pas besoin d'exister au moment de la génération** . Il peut simplement avoir existé dans le tableau à un moment depuis la dernière fois où le tableau a été ré-indexé. L'exemple suivant illustre bien ceci :

La génération automatique de clé peut utiliser un ancien maximum

```
<?php
// Crée un simple tableau
$array = array(1, 2, 3, 4, 5);
print_r($array);

// On efface tous les éléments, mais on conserve le tableau
foreach ($array as $i => $value) {
    unset($array[$i]);
}
print_r($array);

// On ajoute une valeur : notez que la nouvelle clé est 5
// et non pas 0, comme on l'attendrait.
$array[] = 6;
print_r($array);

// Re-indexation :
$array = array_values($array);
$array[] = 7;
print_r($array);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [0] => 1
    [1] => 2
    [2] => 3
    [3] => 4
    [4] => 5
)
Array
(
)
Array
(
    [5] => 6
)
Array
(
    [0] => 6
    [1] => 7
)
```

5.2.6.2 Fonctions pratiques

Il y a toute une panoplie de fonctions pratiques pour travailler avec les tableaux .

Note

La fonction unset permet d'effacer des index dans un tableau. Sachez bien que le tableau ne sera PAS ré-indexé. Si vous utilisez uniquement des index numériques (commençant à zéro et incrémentés de un), vous pouvez effectuer la ré-indexation en utilisant array_values .

Manipuler des éléments de tableau

```
<?php
$a = array(1 => 'un', 2 => 'deux', 3 => 'trois');
unset($a[2]);

/* Cela va produire un tableau qui aurait été
   $a = array(1 => 'un', 3 => 'trois');
   et non pas
   $a = array(1 => 'un', 2 =>'trois');
*/

$b = array_values($a);
// Maintenant b est le tableau array(1 => 'un', 2 =>'trois')
?>
```

L'élément de langage foreach est spécifiquement dédié aux tableaux : il permet de passer en revue simplement les valeurs d'un tableau.

5.2.6.3 A faire et à ne pas faire avec les tableaux

5.2.6.3.1 Pourquoi est-ce que \$foo[bar] est invalide ?

Vous devez toujours utiliser les guillemets autour de la chaîne de caractères placée en tant qu'index du tableau. Par exemple, utilisez \$foo['bar'] et non pas \$foo[bar] . Mais, pourquoi \$foo[bar] est faux ? Vous pourriez avoir vu la syntaxe suivante en vos vieux scripts :

Accès aux index de tableaux, ancienne méthode

```
<?php
$foo[bar] = 'ennemi';
echo $foo[bar];
// etc.
?>
```

Cela est mauvais, mais ça marche. Pourquoi est-ce mauvais? La raison est que PHP attend une constante entre crochets (bar) plutôt qu'une chaîne ('bar', notez les guillemets). Or, PHP pourrait définir dans le futur des constantes qui, malheureusement pour votre code, auront ce nom. En fait, cela fonctionne car la constante indéfinie est transformée en chaîne ayant la même valeur.

Note

Cela ne signifie pas que vous devez **toujours** mettre les clés entre guillemets. Vous n'allez pas utiliser les guillemets avec les clés qui sont des constantes ou des variables , car cela empêchera PHP de les interpréter correctement.

Ne pas mettre les clés de tableaux entre guillemets

```
<?php
error_reporting(E_ALL);
ini_set('display_errors', true);
```

```

ini_set('html_errors', false);
// Tableau simple :
$array = array(1, 2);
$count = count($array);
for ($i = 0; $i < $count; $i++) {
    echo "\nChecking $i : \n";
    echo "Mauvais : " . $array['$i'] . "\n";
    echo "Bon : " . $array[$i] . "\n";
    echo "Mauvais : {$array['$i']}\n";
    echo "Bon : {$array[$i]}\n";
}
?>

```

L'exemple ci-dessus va afficher :

Checking 0:

Notice: Undefined index: \$i in /Users/iboom/Desktop/CVS/docs/mysql/mysql.fr/- on line 10

Mauvais :

Bon : 1

Notice: Undefined index: \$i in /Users/iboom/Desktop/CVS/docs/mysql/mysql.fr/- on line 12

Mauvais :

Bon : 1

Checking 1:

Notice: Undefined index: \$i in /Users/iboom/Desktop/CVS/docs/mysql/mysql.fr/- on line 10

Mauvais :

Bon : 2

Notice: Undefined index: \$i in /Users/iboom/Desktop/CVS/docs/mysql/mysql.fr/- on line 12

Mauvais :

Bon : 2

D'autres exemples :

Exemples de conflits de constantes indéfinies

```

<?php
// Affichons toutes les erreurs
error_reporting(E_ALL);

$arr = array('fruit' => 'pomme', 'legume' => 'carotte');

// Correct
print $arr['fruit']; // pomme
print $arr['legume']; // carotte

// Incorrect. Cela fonctionne mais PHP affiche une alerte car
// il y a utilisation d'une constante indéfinie appelée fruit
//
// Notice: Use of undefined constant fruit - assumed 'fruit' in...
print $arr[fruit]; // pomme

// Définissons une constante pour illustrer le problème.
// Nous créons une constante appelée fruit, qui vaut 'legume'.
define('fruit', 'legume');

// Notez la différence
print $arr['fruit']; // pomme
print $arr[fruit]; // carotte

// La suite est correcte si cela se passe dans une chaîne.
// Les constantes ne sont pas remplacées par leur valeur dans une chaîne

```

```

print "Bonjour $arr[fruit]";      // Bonjour pomme

// A une exception, les accolades entourant un tableau dans une chaîne
// permettent d'utiliser les constantes
print "Bonjour {$arr[fruit]}";    // Bonjour carotte
print "Bonjour {$arr['fruit']}";  // Bonjour pomme

// Ceci ne fonctionnera pas et donnera une erreur d'analyse
// Parse error: parse error, expecting T_STRING' or T_VARIABLE' or T_NUM_STRING'
// Cela s'applique aussi aux autoglobales dans les chaînes
print "Bonjour $arr['fruit']";
print "Bonjour $_GET['foo']";

// La concaténation est une alternative
print "Bonjour " . $arr['fruit']; // Bonjour pomme
?>

```

Lorsque vous activez l' error reporting pour afficher les erreurs de niveau E_NOTICE (en utilisant le niveau E_ALL) alors, vous verrez ces erreurs. Par défaut, error reporting est trop faible pour les afficher.

Comme indiqué dans la section syntaxe , il doit y avoir une expression entre les crochets ('[' et ']'). Cela signifie que vous pouvez écrire ceci :

Utilisation d'une expression dans un index de tableau

```

<?php
echo $arr[somefunc($bar)];
?>

```

Ceci est un exemple d'utilisation de retour de fonction comme index de tableau. PHP reconnaît aussi les constantes, comme la famille des E * .

Utilisation d'une constante dans un index de tableau

```

<?php
$error_descriptions[E_ERROR]    = 'Une erreur fatale est survenue';
$error_descriptions[E_WARNING]  = 'PHP émet une alerte';
$error_descriptions[E_NOTICE]   = 'Ceci est juste une alerte informelle';
?>

```

Notez que E_ERROR est aussi un identifiant valide, tout comme bar dans le premier exemple. Mais le dernier exemple est identique à ceci :

Utilisation d'une constante dans un index de tableau(2)

```

<?php
$error_descriptions[1] = 'Une erreur fatale est survenue';
$error_descriptions[2] = 'PHP émet une alerte';
$error_descriptions[8] = 'Ceci est juste une alerte informationnelle';
?>

```

car E_ERROR vaut 1 , etc.

Comme nous l'avons déjà expliqué dans les exemples ci-dessus, \$foo[bar] fonctionne mais est dangereux. Cela fonctionne car bar est identifié comme une constante. Mais si la constante n'existe pas, PHP suppose que vous souhaitez utiliser bar littéralement, sous la forme "bar" , mais que vous avez oublié les guillemets.

Pourquoi est-ce aussi dangereux?

A l'avenir, le groupe PHP peut décider d'ajouter une constante ou un autre mot clé, ou bien vous souhaitez aussi ajouter une constante dans votre application et vous serez bloqué. Par exemple, vous ne pouvez pas utiliser des mots comme empty et default de cette manière, car ils sont déjà

réservés .

Note

Pour insister, à l'intérieur d'une chaîne de caractères (chaîne de caractères) à guillemets doubles, il est correct de ne pas utiliser les guillemets simples dans les index et donc "\$foo[bar]" est valide. Voyez les exemples ci-dessus pour plus de détails sur la syntaxe des variables dans les chaînes .

5.2.6.4 Conversion en tableau

Pour les types scalaires (entier , nombre à virgule flottante , chaîne de caractères , booléen et ressource), convertir une valeur en tableau retourne un tableau qui contient un élément (à l'index 0) qui est la valeur originale.

Si vous convertissez un tableau en objet, vous obtiendrez des propriétés ou bien cet objet comme élément de tableau. Les clés créées sont les noms des membres.

Si vous convertissez la valeur NULL en tableau, vous obtiendrez un tableau vide.

5.2.6.5 Comparaisons de tableaux

Il est possible de comparer des tableaux grâce à la fonction array_diff et aux opérateurs de tableaux .

5.2.6.6 Exemples

Le type tableau de PHP est très souple. Voici quelques exemples d'utilisation :

Utilisation des tableaux

```
<?php
// ceci
$a = array( 'couleur' => 'rouge',
           'gout' => 'sucre',
           'forme' => 'rond',
           'nom' => 'pomme',
           4 // cette clé sera 0
           );

// est complètement équivalent à
$a['couleur'] = 'rouge';
$a['gout'] = 'sucre';
$a['forme'] = 'rond';
$a['nom'] = 'pomme';
$a[] = 4; // cette clé sera 0

$b[] = 'a';
$b[] = 'b';
$b[] = 'c';
// va créer le tableau array( 0 => 'a' , 1 => 'b' , 2 => 'c' )
// ou plus simplement array('a' , 'b' , 'c' )
?>
```

Utilisation de array

```
<?php
// Array comme correspondance
$map = array( 'version' => 4,
             'OS' => 'Linux',
```

```

        'langue'      => 'français',
        'short_tags' => TRUE
    );
// valeur strictement numériques
$array = array( 7,
               8,
               0,
               156,
               -10
            );
// ceci est la même chose que array( 0 => 7, 1 => 8, ...)
$switching = array( 10, // clé = 0
                   5   => 6,
                   3   => 7,
                   'a' => 4,
                   11, // clé = 6 (index maximum : 5)
                   '8' => 2, // clé = 8 (entier!)
                   '02' => 77, // clé = '02'
                   0   => 12 // la valeur de la clé 10 sera remplacée par 12
                );

// empty array
$empty = array();
?>

```

Collection

```

<?php
$couleurs = array('rouge', 'bleue', 'verte', 'jaune');

foreach ( $couleurs as $couleur ){
    echo "Aimez-vous la couleur $couleur?\n";
}

?>

```

L'exemple ci-dessus va afficher :

```

Aimez-vous la couleur rouge?
Aimez-vous la couleur bleue?
Aimez-vous la couleur verte?
Aimez-vous la couleur jaune?

```

Changer les valeur du tableau directement est possible depuis PHP 5 en les passant par références. Les versions précédentes nécessitent cette solution :

Collection

```

<?php
// PHP 5
foreach ($colors as &$color) {
    $color = strtoupper($color);
}
unset($color); /* s'assure que l'écriture suivante
à $color ne modifie pas le dernier élément du tableau */

// Solution pour les anciennes versions de PHP
foreach ($colors as $key => $color) {
    $colors[$key] = strtoupper($color);
}

print_r($colors);
?>

```

L'exemple ci-dessus va afficher :

```
Array
(
[0] => ROUGE
[1] => BLEU
[2] => VERT
[3] => JAUNE
)
```

Cet exemple crée un tableau d'index minimal 1.

Tableau en 1

```
<?php
$firstquarter = array(1 => 'Janvier', 'Février', 'Mars');
print_r($firstquarter);
```

L'exemple ci-dessus va afficher :

```
Array
(
[1] => 'Janvier'
[2] => 'Février'
[3] => 'Mars'
)
```

Remplissage d'un tableau

```
<?php
// remplit un tableau avec les noms de fichiers d'un dossier
$handle = opendir('.');
while ( $file = readdir($handle) ) {
    $files[] = $file;
}
closedir($handle);
?>
```

Les tableaux sont ordonnés. Vous pouvez modifier l'ordre des valeurs avec de nombreuses fonctions de classement. Voyez les fonctions de [tableaux](#) pour plus d'informations. Vous pouvez compter le nombre d'éléments d'un tableau en utilisant la fonction [count](#) .

Tri de tableaux

```
<?php
sort($files);
print_r($files);
?>
```

Comme une valeur de tableau peut être n'importe quoi, elle peut aussi être un autre tableau. Comme cela, vous pouvez avoir des tableaux multidimensionnels et récursifs.

Tableaux multidimensionnels, et récursifs

```
<?php
$fruits = array ( "fruits" => array ( "a" => "orange",
                                     "b" => "banane",
                                     "c" => "pomme"
                                 ),
                "nombre" => array ( 1,
```

```

                2,
                3,
                4,
                5,
                6,
            ),
            "trou" => array (
                "premier",
                5 => "second",
                "troisième"
            )
        );

// Exemples d'utilisations des tableaux ci-dessus
echo $fruits["trou"][5]; // affiche "second"
echo $fruits["fruits"]["a"]; // affiche "orange"
unset($fruits["trou"][0]); // supprime "premier"

// Créer un tableau multidimensionnel
$jus["pomme"]["vert"] = 'bon';
?>

```

Soyez conscient que l'assignation de valeurs dans un tableau entraîne automatiquement la copie de ces valeurs. Vous devez utiliser l'opérateur de référence pour copier un tableau par référence.

Copie dans un tableau par référence

```

<?php
$arr1 = array(2, 3);
$arr2 = $arr1;
$arr2[] = 4; // $arr2 est modifié,
            // $arr1 vaut toujours array(2, 3)

$arr3 = &$arr1;
$arr3[] = 4; // maintenant $arr1 et $arr3 sont identiques
?>

```

5.2.7 Les objets

5.2.7.1 Initialisation d'un objet

Pour initialiser un objet, vous devez utiliser la commande "new" afin de créer l'instance de l'objet.

Création d'un objet

```

<?php
class foo
{
    function faire_foo()
    {
        echo "Faisant foo.";
    }
}

$bar = new foo;
$bar->faire_foo();
?>

```

Pour une discussion complète, lisez la section sur les [classes et les objets](#) .

5.2.7.2 Conversion en objet

Si un objet est converti en objet, il n'est pas modifié. Si une valeur d'un autre type est convertie en objet, une nouvelle instance de la classe stdClass sera créée. Si la valeur était NULL, la nouvelle instance sera vide. La conversion d'un tableau en objet fera que les propriétés seront nommées par les clés du tableau et avec les valeurs correspondantes. Par toutes les autres valeurs, une variable membre nommé scalar contiendra la valeur.

Transtypage

```
<?php
$obj = (object) 'au revoir';
echo $obj->scalar; // affiche 'au revoir'
?>
```

5.2.8 Ressources

Une ressource ("resource" en anglais), est un type spécial, qui représente une référence sur une ressource externe. Les ressources sont créées par des fonctions dédiées. Reportez-vous à l'annexe [Types des ressources PHP](#) pour une liste exhaustive des fonctions créant et utilisant ces ressources.

Note

Le type de données "resource" a été introduit en PHP 4.

Voir aussi la fonction [get_resource_type](#) .

5.2.8.1 Conversion en ressource

Comme les ressources contiennent des valeurs très spéciales, comme des fichiers ouverts, des connexions aux bases de données, des images ou n'importe quoi d'autre, vous ne pouvez pas convertir une valeur en ressource.

5.2.8.2 Libérer des ressources

Grâce au système de comptabilisation des références introduit en PHP 4 (avec le moteur Zend), PHP détecte automatiquement qu'une ressource n'est plus utilisée (comme Java). Dans ce cas, toutes les ressources systèmes utilisées par cette ressource sont libérées automatiquement. Pour cette raison, il est rarement nécessaire de libérer la mémoire manuellement en utilisant les fonctions free_result associées.

Note

Les connexions persistantes représentent un cas particulier, elles ne seront **PAS** détruites. Voyez la section concernant les [connexions persistantes](#) .

5.2.9 La valeur NULL

La valeur spéciale NULL représente l'absence de valeur. Une variable avec la valeur NULL n'a pas de valeur.

Note

Le type NULL a été introduit en PHP 4.

Une variable est considérée comme NULL si :

- elle a eu d'assigner la constante NULL .
- elle n'a été définie à aucune valeur en fait.
- elle a été détruite par la fonction unset .

5.2.9.1 Syntaxe

Il y a seulement une valeur de type NULL et c'est le mot-clé NULL , insensible à la casse.

La valeur NULL
<pre><?php \$var = NULL; ?></pre>

Voir aussi is_null et unset .

5.2.10 Pseudo-types utilisés dans cette documentation

5.2.10.1 mixed

mixed indique qu'un paramètre accepte plusieurs types, mais pas forcément tous les types.

gettype par exemple, accepte tous les types de variables PHP, mais str_replace n'accepte que des chaînes de caractères et des tableaux.

5.2.10.2 number

number indique qu'un paramètre peut être du type entier ou nombre à virgule flottante .

5.2.10.3 callback

Certaines fonctions comme call_user_function ou usort acceptent une fonction définie par l'utilisateur comme fonction de callback. Les fonctions de callback peuvent être des fonctions simples, mais aussi des objets ou des méthodes statiques de classes.

Une fonction PHP est passée simplement en indiquant son nom, sous forme de chaîne de caractères . Vous pouvez passer une fonction intégrée au langage ou une fonction définie par vos soins, à l'exception de array , echo , empty , eval , exit , isset , list , print et unset .

Une méthode d'un objet instancié est passée sous forme de tableau , contenant un objet comme élément d'index 0 et un nom de méthode comme élément 1 .

Les méthodes statiques de classe peuvent aussi être passées dans l'objet instancié, mais en passant le nom de la classe comme valeur de l'index 0.

Exemple de fonction de callback
<pre><?php // Exemple de fonction de callback function ma_fonction_callback() {</pre>

```

    echo 'Bonjour le monde!';
}

// Exemple de méthode de callback
class MaClasse {
    function MaMethodeCallback() {
        echo 'Bonjour le monde!';
    }
}

// Type 1: Callback simple
call_user_func('ma_fonction_callback');

// Type 2: Appel d'une méthode statique de la classe
call_user_func(array('MyClass', 'MaMethodeCallback'));

// Type 3: Appel d'une méthode objet
$objj = new MaClasse();
call_user_func(array(&$objj, 'MaMethodeCallback'));
?>

```

5.2.11 Définition du type

PHP ne nécessite pas de déclaration explicite du type d'une variable. Le type d'une variable est déterminé par le contexte d'utilisation. Par exemple, si vous assignez une chaîne de caractères à la variable `$var`, `$var` devient une chaîne de caractère. Si vous assignez un nombre entier à `$var`, elle devient un entier.

Un exemple de convertisseur automatique de type est l'opérateur `+`. Si un des opérandes est de type double, alors tous les opérandes sont évalués comme des variables de type double et le résultat est de type double. Sinon, tous les opérandes sont évalués comme des variables de type entier et le résultat sera du type entier. Il est à noter que cela NE CHANGE PAS le type des opérandes. Le seul changement est la manière dont les opérandes sont évalués.

Conversion automatique

```

<?php
$foo = "0"; // $foo est une chaîne de caractères (ASCII 48)
$foo += 2; // $foo est maintenant du type entier (2)
$foo = $foo + 1.3; // $foo est maintenant du type double (3.3)
$foo = 5 + "10 Petits cochons"; // $foo est du type entier (15)
$foo = 5 + "10 cochonnets"; // $foo est du type entier (15)
?>

```

Si les deux derniers exemples vous semblent obscurs ou si vous voulez forcer une variable à être évaluée avec un certain type, reportez-vous au paragraphe "[Conversion de types](#)".

Si vous voulez forcer le type d'une variable, vous pouvez vous reporter à la section concernant le [transtypage](#). Si vous voulez changer le type d'une variable, utilisez la fonction [settype](#).

Note

Le comportement de la conversion automatique en tableau est en fait indéfini.

Conversion automatique indéfinie

```

<?php
$a = '1'; // $a est une chaîne
$a[0] = 'f'; // Mais que sont les index de chaîne?
?>

```

Étant donné que PHP supporte l'indexation de chaîne avec des positions identiques à celles des tableaux, l'exemple ci-dessus conduit à un problème : est ce que \$a est un tableau, dont le premier élément est " f ", ou bien est ce que " f " est le premier élément de la chaîne de caractères \$a ?

Les versions courantes de PHP interprètent la seconde assignation comme un index de chaîne, et donc \$a devient " f " et le résultat de cette conversion automatique doit être considéré comme indéfini. PHP 4 a introduit la syntaxe avec les accolades pour accéder aux caractères d'une chaîne. Utilisez plutôt le code ci-dessous :

Conversion automatique indéfinie(2)

```
<?php
$a    = 'abc'; // $a est une chaîne
$a{1} = 'f';   // $a vaut maintenant "afc"
?>
```

Voyez la section concernant l' [accès aux caractères d'une chaîne](#) pour plus de détails.

5.2.11.1 Transtypage

La conversion de type en PHP fonctionne de la même manière qu'en C : le nom du type désiré est écrit entre parenthèses devant la variable à transtyper ("cast").

Transtypage

```
<?php
$foo = 10; // $foo est un entier
$bar = (double) $foo; // $bar est un double
?>
```

Les conversions autorisées sont :

- (int) , (integer) - type entier
- (bool) , (boolean) - booléen
- (double) , (float) , (real) - type double
- (string) - type chaîne de caractère
- (array) - type tableau
- (object) - type objet

Il est à noter que les tabulations et les espaces sont autorisés à l'intérieur des parenthèses, donc les lignes suivantes sont équivalentes :

Transtypage et espaces

```
<?php
$foo = (int) $bar;
$foo = ( int ) $bar;
?>
```

Note

Au lieu de transtyper une variable en chaîne, vous pouvez aussi l'insérer entre deux guillemets doubles :

Transtypage en chaîne

```
<?php
$foo = 10; // $foo est un entier
$str = "$foo"; // $str est une chaîne
$fst = (string) $foo; // $fst est aussi une chaîne
```

```
// Ceci affiche : "Identique"
if ($fst === $str) {
    echo 'Identique';
}
?>
```

Le transtypage n'a pas toujours un résultat prévisible lors du transtypage entre certains types. Pour plus d'informations, lisez :

- [Conversion en booléen](#)
- [Conversion en entier](#)
- [Conversion en nombre décimal](#)
- [Conversion en chaîne de caractères](#)
- [Conversion en tableau](#)
- [Conversion en objet](#)
- [Conversion en ressource](#)
- [Le tableau de comparaison de types](#)

5.3 Les variables

5.3.1 Essentiel

En PHP, les variables sont représentées par un signe dollar "\$" suivi du nom de la variable. Le nom est sensible à la casse (i.e. \$x != \$X).

Les noms de variables suivent les mêmes règles de nommage que les autres entités PHP. Un nom de variable valide doit commencer par une lettre ou un souligné (_), suivi de lettres, chiffres ou soulignés. Exprimé sous la forme d'une expression régulière, cela donne : '[\[a-zA-Z_\x7f-\xff\]\[a-zA-Z0-9_\x7f-\xff\]*](#)'

Note

Dans nos propos, une lettre peut être une des lettres minuscules (a à z) ou majuscules (A à Z) et les caractères ASCII de 127 à 255 (0x7f-0xff).

Pour obtenir des informations sur une variable, voyez les fonctions dédiées aux [variables](#) .

Validité des noms de variables

```
<?php
$var = 'Jean';
$Var = 'Paul';
echo "$var, $Var";           // affiche "Jean, Paul"

$4site = 'pas encore';     // invalide : commence par un nombre
$_4site = 'pas encore';    // valide : commence par un souligné
$täyte = 'mansikka';       // valide : 'ä' est ASCII (étendu) 228.
?>
```

En PHP 3, les variables sont toujours assignées par valeur. C'est-à-dire, lorsque vous assignez une expression à une variable, la valeur de l'expression est copiée dans la variable. Cela signifie, par exemple, qu'après avoir assigné la valeur d'une variable à une autre, modifier l'une des variables n'aura pas d'effet sur l'autre. Pour plus de détails sur ce genre d'assignation, reportez-vous aux [expressions](#) .

Depuis PHP 4, PHP permet aussi d'assigner les valeurs aux variables par référence . Cela signifie que la nouvelle variable ne fait que référencer (en d'autres terme, "devient un alias de", ou encore "pointe sur") la variable originale. Les modifications de la nouvelle variable affecteront l'ancienne et vice versa.

Pour assigner par référence, ajoutez simplement un & (ET commercial) au début de la variable qui est assignée (la variable source). Dans l'exemple suivant, Mon nom est Pierre s'affichera deux fois :

Assignation de référence

```
<?php
$foo = 'Pierre';           // Assigne la valeur 'Pierre' à $foo
$bar = &$foo;             // Référence $foo avec $bar.
$bar = "Mon nom est $bar"; // Modifie $bar...
echo $foo;                // $foo est aussi modifiée
echo $bar;
?>
```

Une chose importante à noter est que seules les variables nommées peuvent être assignées par référence.

Assignation de référence et variables anonymes

```
<?php
$foo = 25;
$bar = &$foo;           // assignation valide
$bar = &(24 * 7);      // assignation invalide : référence une expression sans nom
function test() {
    return 25;
}
$bar = &test();        // assignation invalide.
?>
```

Il n'est pas nécessaire d'initialiser les variables en PHP, cependant, cela reste une excellente pratique. Les variables non initialisées ont une valeur par défaut selon leurs types - FALSE , zéro, chaîne vide ou un tableau vide.

Valeurs par défaut des variables non initialisées

```
<?php
echo ($unset_bool ? "true" : "false"); // false
$unset_int += 25; // 0 + 25 => 25
echo $unset_string . "abc"; // "" . "abc" => "abc"
$unset_array[3] = "def"; // array() + array(3 => "def") => array(3 => "def")
?>
```

Utiliser la valeur par défaut d'une variable non initialisée est problématique lorsque vous incluez un fichier dans un autre qui utilise le même nom de variable. C'est également un risque niveau sécurité lorsque register_globals est activé. Une erreur de niveau E_NOTICE sera émise lorsque vous travaillerez avec des variables non initialisées, cependant, aucune erreur ne sera lancée lorsque vous tenterez d'insérer un élément dans un tableau non initialisé. La structure de langage isset peut être utilisée pour détecter si une variable a déjà été initialisée.

5.3.2 Variables pré-définies

PHP fourni un grand nombre de variables pré-définies. Cependant, beaucoup de ces variables ne peuvent pas être présentées ici, car elles dépendent du serveur sur lequel elles tournent, de la version du serveur et de la configuration du serveur ou encore d'autres facteurs. Certaines de ces variables ne seront pas accessibles lorsque PHP fonctionne en exécutable.

Attention

Depuis la version PHP 4.2.0, la valeur par défaut de la directive PHP `register_globals` est **off** . Ceci est une évolution majeure de PHP. Avoir la directive `register_globals` à **off** affecte les variables pré-définies du contexte globale. Par exemple, pour lire `DOCUMENT_ROOT` vous devez utiliser `$_SERVER['DOCUMENT_ROOT']` au lieu de `$DOCUMENT_ROOT` ou bien, il faut lire `$_GET['id']` dans l'URL `http://www.example.com/test.php?id=3` au lieu de `$id` ou encore `$_ENV['HOME']` au lieu de `$HOME` .

Pour des informations liées à cette évolution, lisez la documentation de la directive `register_globals` , le chapitre sur la sécurité, à propos de l' [Utilisation des variables superglobales](#) , ainsi que les annonces de PHP [4.1.0](#) et [4.2.0](#) .

L'utilisation des variables pré-définies de PHP, comme les [tableaux superglobaux](#) , est recommandé.

Depuis la version 4.1.0, PHP fournit un jeu de tableaux pré-définis, contenant les variables du serveur (si possible), les variables d'environnement et celle d'entrées. Ces nouveaux tableaux sont un peu particuliers, car ils sont automatiquement globaux : ils sont automatiquement disponibles dans tous les environnements d'exécution, sans avoir à utiliser le mot réservé `global` . Pour cette raison, ils sont dits 'auto-globaux' ou bien encore 'superglobaux' (il n'y a pas de mécanisme PHP pour créer de telles variables. Les superglobales sont listées ci-dessous. Cependant, pour connaître le détails de leur contenu et une présentation approfondie sur les variables pré-définies PHP et leur nature, reportez-vous à la section [variables pré-définies](#) . De plus, vous noterez que les anciennes variables pré-définies (`$HTTP_*_VARS`) existent toujours. Depuis PHP 5.0.0, les [tableaux prédéfinis](#) PHP peuvent être désactivés avec l'option de configuration `register_long_arrays` .

Note**Variables variables**

Les superglobales ne peuvent pas être utilisées comme [variables dynamiques](#) dans les fonctions ou les méthodes des classes.

Note

Même si les superglobales et `HTTP_*_VARS` peuvent exister en même temps, ils ne sont pas identiques, donc, le changement d'une ne changera pas l'autre.

Si certaines variables de [variables_order](#) ne sont pas définies, leur tableau pré-défini PHP correspondant est laissé vide.

Tableaux superglobaux de PHP**\$GLOBALS**

Contient une référence sur chaque variable qui est en fait disponible dans l'environnement d'exécution global. Les clés de ce tableau sont les noms des variables globales. `$GLOBALS` existe depuis PHP 3.

\$_SERVER

Les variables fournies par le serveur web, ou bien directement liées à l'environnement d'exécution du script courant. C'est la nouvelle version de l'ancienne variable `$HTTP_SERVER_VARS` , qui est maintenant obsolète, mais toujours là.

\$_GET

Les variables fournies au script via la chaîne de requête URL. C'est la nouvelle version de l'ancienne variable `$HTTP_GET_VARS` qui est maintenant obsolète, mais toujours là.

\$_POST

Les variables fournies par le protocole HTTP en méthode POST. C'est la nouvelle version de l'ancienne variable \$HTTP_POST_VARS qui est maintenant obsolète, mais toujours là.

\$ _COOKIE

Les variables fournies par le protocole HTTP, dans les cookies. C'est la nouvelle version de l'ancienne variable \$HTTP_COOKIE_VARS qui est maintenant obsolète, mais toujours là.

\$ _FILES

Les variables fournies par le protocole HTTP, suite à un téléchargement de fichier. C'est la nouvelle version de l'ancienne variable \$HTTP_POST_FILES qui est maintenant obsolète, mais toujours là. Voir Téléchargement par méthode POST , pour plus d'informations.

\$ _ENV

Les variables fournies par l'environnement. C'est la nouvelle version de l'ancienne variable \$HTTP_ENV_VARS qui est maintenant obsolète, mais toujours là.

\$ _REQUEST

Les variables fournies au script par n'importe quel mécanisme d'entrée et qui ne doit recevoir une confiance limitée. Note : lorsque vous exécutez un script en ligne de commande, cette variable ne va **pas** inclure les variables argv et argc . Elles seront présentes dans la variable \$_SERVER . La présence et la valeur des entrées de ce tableau sont réglés par la directive variables_order . Ce tableau n'est l'évolution d'aucune variable d'avant PHP 4.1.0.

Attention
Depuis PHP 4.3.0, les informations de la variable \$_FILES ne sont plus intégrées dans la variable \$_REQUEST .
Note
En <u>ligne de commande</u> , cette variable n'inclut pas les variables argv et argc : elles sont stockées dans le tableau \$_SERVER .

\$ _SESSION

Les variables qui sont en fait enregistrées dans la session attachée au script. C'est la nouvelle version de l'ancienne variable \$HTTP_SESSION_VARS . Voir le chapitre sur la gestion des sessions pour plus d'informations.

5.3.3 Portée des variables

La portée d'une variable dépend du contexte dans lequel la variable est définie. Pour la majorité des variables, la portée concerne la totalité d'un script PHP. Mais lorsque vous définissez une fonction, la portée d'une variable définie dans cette fonction est locale à la fonction. Par exemple :

Les variables sont locales à la fonction
<pre><?php \$a = 1; include 'b.inc'; ?></pre>

Ici, la variable \$a sera accessible dans le script inclus b.inc . Cependant, dans les fonctions définies par l'utilisateur, une nouvelle définition de cette variable sera donnée, limitée à la fonction. Toute variable utilisée dans une fonction est par définition, locale. Par exemple :

Les variables sont locales à la fonction (2)
<pre><?php \$a = 1; /* portée globale */</pre>

```
function test() {
    echo $a; /* portée locale */
}
test();
?>
```

Le script n'affichera rien à l'écran car l'instruction `echo` utilise la variable locale `$a`, et celle-ci n'a pas été assignée préalablement dans la fonction. Vous pouvez noter que ce concept diffère un petit peu du langage C dans lequel une variable globale est automatiquement accessible dans les fonctions, à moins d'être redéfinie localement dans la fonction. Cela peut poser des problèmes si vous redéfinissez des variables globales localement. En PHP, une variable globale doit être déclarée à l'intérieur de chaque fonction afin de pouvoir être utilisée dans cette fonction.

5.3.3.1 Le mot clé global

Commençons par un exemple avec `global` :

Exemple avec global

```
<?php
$a = 1;
$b = 2;
function somme() {
    global $a, $b;
    $b = $a + $b;
}
somme();
echo $b;
```

Le script ci-dessus va afficher la valeur 3. En déclarant globales les variables `$a` et `$b` locales de la fonction `somme()`, toutes les références à ces variables concerneront les variables globales. Il n'y a aucune limite au nombre de variables globales qui peuvent être manipulées par une fonction.

Une deuxième méthode pour accéder aux variables globales est d'utiliser le tableau associatif pré-défini `$GLOBALS`. Le précédent exemple peut être réécrit de la manière suivante :

Les variables globales et \$GLOBALS

```
<?php
$a = 1;
$b = 2;
function somme() {
    $GLOBALS['b'] = $GLOBALS['a'] + $GLOBALS['b'];
}
somme();
echo $b;
?>
```

Le tableau `$GLOBALS` est un tableau associatif avec le nom des variables globales comme clé et les valeurs des éléments du tableau comme valeur des variables. Notez que `$GLOBALS` existe dans tous les contextes, car `$GLOBALS` est un superglobal. Voici un exemple des super globaux :

Les variables super globales

```
<?php
function test_global() {

    // La plupart des variables pré-définies ne sont pas des "superglobales" et
    // requiert le mot-clé 'global' pour être disponible dans une fonction.
    global $HTTP_POST_VARS;
```

```

echo $HTTP_POST_VARS['name'];

// Les superglobales sont accessibles dans tous les contextes
// et ne requièrent pas 'global'. Les superglobales sont disponibles
// depuis PHP 4.1.0 et HTTP_POST_VARS est de plus en plus
// déprécié.
echo $_POST['name'];
}
?>

```

5.3.3.2 Utilisation des variables static

Une autre caractéristique importante de la portée des variables est la notion de variable static . Une variable statique a une portée locale uniquement, mais elle ne perd pas sa valeur lorsque le script appelle la fonction. Prenons l'exemple suivant :

Les variables statiques

```

<?php
function Test()
{
    $a = 0;
    echo $a;
    $a++;
}
?>

```

Cette fonction est un peu inutile car à chaque fois qu'elle est appelée, elle initialise \$a à 0 et affiche "0". L'incrémement de la variable (\$a ++) ne sert pas à grand chose, car dès que la fonction est terminée, la variable disparaît. Pour faire une fonction de comptage utile, c'est-à-dire qui ne perdra pas la trace du compteur, la variable \$a est déclarée comme une variable statique :

Les variables statiques (2)

```

<?php
function Test() {
    static $a = 0;
    echo $a;
    $a++;
}
?>

```

Maintenant, à chaque fois que la fonction Test() est appelée, elle affichera une valeur de \$a incrémentée de 1.

Les variables statiques sont essentielles lorsque vous faites des appels récursifs à une fonction. Une fonction récursive est une fonction qui s'appelle elle-même. Il faut faire attention lorsque vous écrivez une fonction récursive car il est facile de faire une boucle infinie. Vous devez vérifier que vous avez bien une condition qui permet de terminer votre récursivité. La fonction suivante compte récursivement jusqu'à 10 :

Les variables statiques et la récursivité

```

<?php
function Test()
{
    static $count = 0;
    $count++;
    echo $count;
    if ($count < 10) {
        Test();
    }
}

```

```

    $count--;
}
?>

```

Note

Les variables statiques doivent être déclarées comme dans l'exemple ci-dessus. Tenter d'assigner des valeurs à ces variables qui sont le résultat d'expressions causera une erreur d'analyse.

Déclaration de variables statiques

```

<?php
function foo(){
    static $int = 0;           // correct
    static $int = 1+2;        // faux  (car c'est une expression)
    static $int = sqrt(121);  // faux  (car c'est aussi une expression)

    $int++;
    echo $int;
}
?>

```

5.3.3.3 Les références avec les variables global et static

Le Zend Engine 1, sur qui repose PHP 4, implémente les options `static` et `global` pour les variables, en terme de références. Par exemple, une vraie variable globale est importée dans un contexte de fonction avec `global`. Cette commande crée en fait une référence sur la variable globale. Cela peut vous mener à des comportements inattendus, par exemple :

Les variables statiques et les références

```

<?php
function test_global_ref() {
    global $obj;
    $obj = &new stdClass;
}

function test_global_noref() {
    global $obj;
    $obj = new stdClass;
}

test_global_ref();
var_dump($obj);
test_global_noref();
var_dump($obj);
?>

```

L'exemple ci-dessus va afficher :

```

NULL
object(stdClass)(0) {
}

```

Un comportement similaire s'applique à la commande `static`. Les références ne sont pas stockées dynamiquement :

Les variables statiques et les références (2)

```

<?php
function &get_instance_ref() {

```

```

static $obj;

echo 'Objet statique : ';
var_dump($obj);
if (!isset($obj)) {
    // Assigne une référence à une variable statique
    $obj = &new stdClass;
}
$obj->property++;
return $obj;
}

function &get_instance_noref() {
    static $obj;

    echo 'Objet statique : ';
    var_dump($obj);
    if (!isset($obj)) {
        // Assigne une objet à une variable statique
        $obj = new stdClass;
    }
    $obj->property++;
    return $obj;
}

$obj1 = get_instance_ref();
$still_obj1 = get_instance_ref();
echo "\n";
$obj2 = get_instance_noref();
$still_obj2 = get_instance_noref();
?>

```

L'exemple ci-dessus va afficher :

Objet statique : NULL

Objet statique : NULL

Objet statique : NULL

Objet statique : object(stdClass)(1) {

["property"]=>

int(1)

}

Ces exemples illustrent les problèmes rencontrés lors de l'assignation de référence à des variables statiques, qui sont **oubliées** lorsque vous appelez &get_instance_ref() une seconde fois.

5.3.4 Les variables dynamiques

Il est pratique d'avoir parfois des noms de variables qui sont variables. C'est-à-dire un nom de variable qui est affecté et utilisé dynamiquement. Une variable classique est affectée avec l'instruction suivante :

Une variable classique

```

<?php
$a = 'bonjour';
?>

```

Une variable dynamique prend la valeur d'une variable et l'utilise comme nom d'une autre variable. Dans l'exemple ci-dessous, **bonjour** peut être utilisé comme le nom d'une variable en utilisant le "\$\$" précédant la variable. C'est-à-dire :

Les variables variables

```
<?php
$$a = 'monde';
?>
```

A ce niveau, deux variables ont été définies et stockées dans l'arbre des symboles PHP : \$a avec comme valeur "bonjour" et \$bonjour avec comme valeur "monde". Alors, l'instruction :

Equivalent de variables variables

```
<?php
echo "$a ${$a}";
?>
```

produira le même affichage que :

Equivalent de variables variables

```
<?php
echo "$a $bonjour";
?>
```

c'est-à-dire : **bonjour monde** .

Afin de pouvoir utiliser les variables dynamiques avec les tableaux, vous avez à résoudre un problème ambigu. Si vous écrivez \$\$a[1] , l'analyseur a besoin de savoir si vous parlez de la variable qui a pour nom \$a[1] ou bien si vous voulez l'index [1] de la variable \$\$a . La syntaxe pour résoudre cette ambiguïté est la suivante : \${\$a[1]} pour le premier cas et \${\$a}[1] pour le deuxième.

Attention

Notez que les variables dynamiques ne peuvent pas être utilisées avec les [tableaux Superglobaux](#) de PHP.

5.3.5 Variables externes à PHP

5.3.5.1 Formulaires HTML (GET et POST)

Lorsqu'un formulaire est envoyé à un script PHP, toutes les variables du formulaire seront automatiquement disponibles dans le script. Par exemple, considérons le formulaire suivant :

Exemple avec un formulaire simple

```
<form action="foo.php" method="post">
  Nom : <input type="text" name="username" /><br />
  Email: <input type="text" name="email" /><br />
  <input type="submit" name="submit" value="Envoi!" />
</form>
```

Suivant votre configuration particulière et vos préférences, vous avez plusieurs méthodes pour accéder aux variables du formulaire. Voici quelques exemples :

Accéder simplement à des variables de formulaires POST

```
<?php
```

```
// Disponibles depuis PHP 4.1.0

echo $_POST['username'];
echo $_REQUEST['username'];

import_request_variables('p', 'p_');
echo $p_username;

// Disponibles depuis PHP 3.

echo $HTTP_POST_VARS['username'];

// Disponibles si la directive register_globals = on. Depuis
// PHP 4.2.0 la valeur par défaut de cette directive est register_globals = off.
// Utiliser ou présumer cette méthode est découragé.

echo $username;
?>
```

Utiliser un formulaire de type GET est similaire, hormis le fait que vous deviez utiliser les variables pré-définies de GET à la place. GET s'applique aussi à la QUERY_STRING (les informations disponibles après le '?' dans une URL). De ce fait, par exemple, <http://www.example.com/test.php?id=3> contient les données de GET, qui sont accessibles via `$_GET['id']`. Voyez aussi [\\$_REQUEST](#) et [import_request_variables](#).

Note

Les tableaux superglobaux, comme `$_POST` et `$_GET` sont disponibles depuis PHP 4.1.0.

Comme nous l'avons déjà dit, avant PHP 4.2.0, la valeur par défaut de `register_globals` était **on**. Dans PHP 3, elle était toujours à On. La communauté PHP n'encourage personne à utiliser cette directive et privilégie la valeur **off** et un code accordé.

Note

La directive de configuration `magic_quotes_gpc` affecte les valeurs de GET, POST et cookies. Si elle est activée, une valeur comme celle de (C'est "PHP!") sera magiquement transformée en (C'est \"PHP!\"). La protection des caractères est nécessaire pour l'insertion dans les bases de données. Voyez aussi les fonctions [addslashes](#), [stripslashes](#) et [magic_quotes_sybase](#).

PHP comprend aussi les tableaux dans le contexte des formulaires. (voir aussi [la FAQ](#)). Vous pouvez, par exemple, grouper des variables ensembles ou bien utiliser cette fonctionnalité pour lire des valeurs multiples d'un menu déroulant. Par exemple, voici un formulaire qui se poste lui-même des données, et les affiche :

Variables de formulaires complexes

```
<?php
if ($_POST['action'] == 'submitted') {
    echo '<pre>';

    print_r($_POST);
    echo '<a href="'. $_SERVER['PHP_SELF'] .' ">Essayez à nouveau</a>';

    echo '</pre>';
} else {
?>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
    Nom : <input type="text" name="personal[name]" /><br />
    Email : <input type="text" name="personal[email]" /><br />
    Biere : <br />
    <select multiple name="vin[]">
        <option value="bordeaux">bordeaux</option>
        <option value="beaujolais">beaujolais</option>
        <option value="loire">loire</option>
```

```

</select><br />
<input type="hidden" name="action" value="submitted" />
<input type="submit" name="submit" value="submit me!" />
</form>
<?php
}
?>

```

En PHP 3, la syntaxe des tableaux HTML est limitée aux tableaux uni-dimensionnels. En PHP 4, ces limitations ont été levées.

5.3.5.1.1 Nom de variables IMAGE de type SUBMIT

Lors de la soumission d'un formulaire, il est possible d'utiliser une image au lieu d'un bouton standard, comme ceci :

Utilisation d'une image pour soumettre un formulaire

```
<input type="image" src="image.gif" name="sub" />
```

Lorsque l'utilisateur clique sur cette image, le formulaire associé est envoyé au serveur, avec deux données supplémentaires, `sub_x` et `sub_y`. Elles contiennent les coordonnées du clic de l'utilisateur dans l'image. Vous noterez que ces variables sont envoyées par le navigateur avec un point dans leur nom, mais PHP convertit ces points en soulignés.

5.3.5.2 Cookies HTTP

PHP supporte les cookies HTTP de manière totalement transparente, comme défini dans les [spécifications de Netscape](#). Les cookies sont un mécanisme permettant de stocker des données sur la machine cliente à des fins d'identification de l'utilisateur. Vous pouvez établir un cookie grâce à la fonction [setcookie](#). Les cookies font partie intégrante des en-têtes HTTP et donc la fonction [setcookie](#) doit être appelée avant que le moindre affichage ne soit envoyé au navigateur. C'est la même restriction que pour la fonction [header](#). Les données contenues dans les cookies sont alors disponibles dans les tableaux de cookies appropriés, comme `$_COOKIE`, `$HTTP_COOKIE_VARS` mais aussi `$_REQUEST`. Lisez la page de la documentation sur la fonction [setcookie](#) pour plus de détails ainsi que des exemples.

Si vous souhaitez assigner plusieurs valeurs à un seul cookie, il vous faut ajouter les caractères `[]` au nom de votre cookie. Par exemple :

Utilisation de tableaux avec les cookies

```

<?php
setcookie("MyCookie[foo]", 'Test 1', time()+3600);
setcookie("MyCookie[bar]", 'Test 2', time()+3600);
?>

```

Cela va créer deux cookies distinctes bien que `MyCookie` est maintenant un simple tableau dans votre script. Si vous voulez définir seulement un cookie avec plusieurs valeurs, utilisez la fonction [serialize](#) ou [explode](#) sur la première valeur.

Il est à noter qu'un cookie remplace le cookie précédent par un cookie de même nom tant que le "path" ou le domaine sont identiques. Donc, pour une application de panier, vous devez implémenter un compteur et l'incrémenter au fur et à mesure. C'est-à-dire :

Exemple avec [setcookie](#)

```
<?php
```

```

if (isset($_COOKIE['compte'])) {
    $compte = $_COOKIE['compte'] + 1;
} else {
    $compte = 1;
}
setcookie('Panier', $compte, time()+3600);
setcookie("Panier[$compte]", $item, time()+3600);
?>

```

5.3.5.3 Cas des points dans les noms de variables

Typiquement, PHP ne modifie pas les noms des variables lorsqu'elles sont passées à un script. Cependant, il faut noter que les points (.) ne sont pas autorisés dans les noms de variables PHP. Pour cette raison, jetez un oeil sur :

```

<?php
    $varname.ext; /* nom de variable invalide */
?>

```

Dans ce cas, l'analyseur croit voir la variable nommée \$varname , suivie par l'opérateur de concaténation, et suivie encore par la chaîne sans guillemets (une chaîne sans guillemets et qui n'a pas de signification particulière). Visiblement, ce n'est pas ce qu'on attendait...

Pour cette raison, il est important de noter que PHP remplacera automatiquement les points des noms de variables entrantes par des soulignés (underscore).

5.3.5.4 Détermination du type des variables

Parce que PHP détermine le type des variables et les convertit (généralement) comme il faut, ce n'est pas toujours le type de variable que vous souhaitez. PHP inclut des fonctions permettant de déterminer le type d'une variable : gettype , is_array , is_float , is_int , is_object et is_string . Lisez également le chapitre sur les types .

5.4 Les constantes

Une constante est un identifiant (un nom) qui représente une valeur simple. Comme son nom le suggère, cette valeur ne peut jamais être modifiée durant l'exécution du script (les constantes magiques `__FILE__` et `__LINE__` sont les seules exceptions). Le nom d'une constante est sensible à la casse, par défaut. Par convention, les constantes sont toujours en majuscules.

Les noms de constantes suivent les mêmes règles que n'importe quel nom en PHP. Un nom de constante valide commence par une lettre ou un souligné (`_`), suivi d'un nombre quelconque de lettre, chiffres ou soulignés. Sous forme d'expression régulière, cela peut s'exprimer comme ceci : `[a-zA-Z_\x7f-\xff][a-zA-Z0-9_\x7f-\xff]*`

Noms valides et invalides pour les constantes

```

<?php
// Noms valides
define("FOO",      "something");
define("FOO2",     "something else");
define("FOO_BAR",  "something more");

// Noms invalides

```

```
define("2FOO", "something");

// Ce nom est valide, mais évitez-le:
// PHP peut un jour fournir une constante magique nommée
// ainsi, ce qui va corrompre vos scripts.
define("__FOO__", "something");

?>
```

Note

Dans cette documentation, une lettre peut être un des caractères suivants : de a à z, de A à Z et tous les caractères ASCII de 127 à 255 (0x7f-0xff).

Tous comme les [superglobals](#), les constantes sont accessibles de manière globale. Vous pouvez la définir n'importe où, et y accéder depuis n'importe quelle fonction.

5.4.1 Syntaxe

Vous pouvez définir une constante en utilisant la fonction [define](#). Une fois qu'une constante est définie, elle ne peut jamais être modifiée, ou détruite.

Seuls les types de données scalaires peuvent être placés dans une constante : c'est à dire les types booléen, entier, double et chaîne de caractères (soit bool, [entier](#), double et string).

Vous pouvez accéder à la valeur d'une constante en spécifiant simplement son nom. Contrairement aux variables, vous ne devez **PAS** préfixer le nom de la constante avec \$. Vous pouvez aussi utiliser la fonction [constant](#), pour lire dynamiquement la valeur d'une constante, dont vous obtenez le nom dynamiquement (retour de fonction, par exemple). Utilisez la fonction [get_defined_constants](#) pour connaître la liste de toutes les constantes définies.

Note

Les constantes et les variables globales utilisent deux espaces de noms différents. Ce qui implique que TRUE et \$TRUE sont généralement différents (en tous cas, ils peuvent avoir des valeurs différentes).

Lorsque vous utilisez une constante non définie, PHP suppose que vous utilisez le nom de la constante. Une [note](#) sera générée. Utilisez la fonction [defined](#) pour savoir si une constante existe ou pas.

Il y a des différences entre les constantes et les variables :

- Les constantes ne commencent pas par le signe (\$);
- Les constantes sont définies et accessibles à tout endroit du code, globalement.
- Les constantes ne peuvent pas être redéfinies ou indéfinies une fois qu'elles ont été définies.
- Les constantes ne peuvent contenir que des scalaires.

Définir une constante

```
<?php
define("CONSTANTE", "Bonjour le monde.");
echo CONSTANTE; // affiche "Bonjour le monde."
echo Constante; // affiche "Constante" et une note.
?>
```

Voir aussi les [constantes de classe](#).

5.4.2 Constantes magiques

PHP fournit un grand nombre de constantes magiques . Certaines constantes sont définies par différentes extensions, et ne seront présentes que si ces extensions sont compilées avec PHP, ou bien si l'extension a été chargée dynamiquement.

Il y a cinq constantes magiques qui changent suivant l'emplacement où elles sont utilisées. Par exemple, la valeur de `__LINE__` dépend de la ligne où vous l'utilisez dans votre script. Ces constantes spéciales sont insensibles à la casse.

Nom	Description
<code>__LINE__</code>	La ligne courante dans le fichier.
<code>__FILE__</code>	Le chemin complet et le nom du fichier courant. Si utilisé dans un include, le nom du fichier inclus est retourné. Depuis PHP 4.0.2, <code>__FILE__</code> contient toujours le chemin absolu alors que les anciennes versions contenaient le chemin relatif, dans certaines circonstances.
<code>__FUNCTION__</code>	Le nom de la fonction. (Ajouté en PHP 4.3.0) Depuis PHP 5, cette constante retourne le nom de la fonction comme il a été déclaré (sensible à la casse). En PHP 4, cette valeur est toujours en minuscule.
<code>__CLASS__</code>	Le nom de la classe courante. (Ajouté en PHP 4.3.0) Depuis PHP 5, cette constante retourne le nom de la classe comme il a été déclaré (sensible à la casse). En PHP 4, cette valeur est toujours en minuscule.
<code>__METHOD__</code>	Le nom de la méthode courante. (Ajouté en PHP 5.0.0) Le nom de la méthode est retourné comme il a été déclaré (sensible à la casse).

Voir aussi [get_class](#) , [get_object_vars](#) , [file_exists](#) et [function_exists](#) .

5.5 Les expressions

Les expressions sont la partie la plus importante de PHP. En PHP, presque tout ce que vous écrivez est une expression. La manière la plus simple de définir une expression est : "tout ce qui a une valeur".

Les formes les plus simples d'expressions sont les constantes et les variables. Lorsque vous écrivez " `$a = 5`", vous assignez la valeur '5' à la variable `$a` . Bien évidemment, '5' vaut 5 ou, en d'autres termes, '5' est une expression avec pour valeur 5 (dans ce cas, '5' est un entier constant).

Après cette assignation, vous pouvez considérer que `$a` a pour valeur 5 et donc, écrire `$b = $a` , revient à écrire `$b = 5`. En d'autres termes, `$a` est une expression avec une valeur de 5. Si tout fonctionne correctement, c'est exactement ce qui arrive.

Un exemple plus complexe concerne les fonctions. Par exemple, considérons la fonction suivante :

Affectation d'expressions
<pre><?php function foo () { return 5; } ?></pre>

En supposant que vous êtes familiers avec le concept de fonction, (si ce n'est pas le cas, jetez un oeil au chapitre concernant les [fonctions](#)), vous serez d'accord que `$c = foo()` est équivalent à `$c = 5` , et vous aurez tout à fait raison. Les fonctions sont des expressions qui ont la valeur de leur

"valeur de retour". Si `foo()` renvoie 5, la valeur de l'expression ' `foo()` ' est 5. Habituellement, les fonctions ne font pas que renvoyer une valeur constante mais réalisent des traitements.

Bien sûr, les valeurs en PHP n'ont pas à être des valeurs numériques, comme c'est souvent le cas. PHP supporte quatre types de variables scalaires : les valeurs entières (entier), les nombres à virgule flottante (float), les chaînes de caractères (chaîne de caractères) et les booléens (boolean). (une variable scalaire est une variable que vous ne pouvez pas scinder en morceaux, au contraire des tableaux par exemple). PHP supporte aussi deux types composés : les tableaux et les objets. Chacun de ces types de variables peut être affecté ou renvoyé par une fonction.

Les utilisateurs de PHP/FI 2 ne verront aucun changement. Malgré tout, PHP va plus loin dans la gestion des expressions, comme le font d'autres langages. PHP est un langage orienté expression, dans le sens où presque tout est une expression. Considérons l'exemple dont nous avons déjà parlé, ' `$a = 5` '. Il est facile de voir qu'il y a deux valeurs qui entrent en jeu ici, la valeur numérique constante '5' et la valeur de la variable `$a` qui est mise à jour à la valeur 5. Mais, la vérité est qu'il y a une autre valeur qui entre en jeu ici et c'est la valeur de l'assignation elle-même. L'assignation elle-même est assignée à une valeur, dans ce cas-là 5. En pratique, cela signifie que ' `$a = 5` ' est une expression qui a pour valeur 5. Donc, écrire ' `$b = ($a = 5)` ' revient à écrire ' `$a = 5; $b = 5;` ' (un point virgule marque la fin d'une instruction). Comme les assignations sont analysées de droite à gauche, vous pouvez aussi bien écrire ' `$b = $a = 5` '.

Un autre bon exemple du langage orienté expression est la pré-incrémentation et la post-incrémentation, (ainsi que la décrémentation). Les utilisateurs de PHP/FI 2 et ceux de nombreux autres langages sont habitués à la notation " `variable++` " et " `variable--` ". Ce sont les opérateurs d'incrément et de décrémentation . En PHP/FI 2, l'instruction ' `$a ++` ' n'a aucune valeur (c'est-à-dire que ce n'est pas une expression) et vous ne pouvez donc pas l'utiliser. PHP ajoute les possibilités d'incrément et de décrémentation comme c'est le cas dans le langage C. En PHP, comme en C, il y a deux types d'opérateurs d'incrément (pré-incrémentation et post-incrémentation). Les deux types d'opérateur d'incrément jouent le même rôle (c'est-à-dire qu'ils incrémentent la variable). La différence vient de la valeur de l'opérateur d'incrément. L'opérateur de pré-incrémentation, qui s'écrit ' `++ $variable` ', évalue la valeur incrémentée (PHP incrémente la variable avant de lire la valeur de cette variable, d'où le nom de pré-incrémentation). L'opérateur de post-incrémentation, qui s'écrit ' `$variable ++` ', évalue la valeur de la variable avant de l'incrémenter (PHP incrémente la variable après avoir lu sa valeur, d'où le nom de post-incrémentation).

Un type d'expression très commun est l'expression de comparaison. Ces expressions sont évaluées à 0 ou 1, autrement dit FALSE ou TRUE , respectivement. PHP supporte les opérateurs de comparaison > (plus grand que), => (plus grand ou égal), == (égal à), < (plus petit que), <= (plus petit ou égal). Ces expressions sont utilisées de manière courante dans les instructions conditionnelles, comme l'instruction `if` .

Pour le dernier exemple d'expression, nous allons parler des combinaisons d'opérateurs/assignation. Vous savez que si vous voulez incrémenter la variable `$a` d'une unité, vous devez simplement écrire ' `$a ++` '. Mais si vous voulez ajouter la valeur '3' à votre variable ? Vous pouvez écrire plusieurs fois ' `$a ++` ', mais ce n'est pas la meilleure des méthodes. Une pratique plus courante est d'écrire ' `$a = $a + 3` '. L'expression ' `$a + 3` ' correspond à la valeur `$a` plus 3, et est de nouveau assignée à la variable `$a` . Donc, le résultat est l'incrément de 3 unités. En PHP, comme dans de nombreux autres langages comme le C, vous pouvez écrire cela de manière plus concise, manière qui avec le temps se révélera plus claire et plus rapide à comprendre. Ajouter 3 à la valeur de la variable `$a` peut s'écrire ' `$a += 3` '. Cela signifie précisément : "on prend la valeur de la variable `$a` , on ajoute la valeur 3 et on assigne cette valeur à la variable `$a` ". Et pour être plus concis et plus clair, cette expression est plus rapide. La valeur de l'expression ' `$a += 3` ', comme l'assignation d'une valeur quelconque, est la valeur assignée. Il est à noter que ce n'est pas

3 mais la combinaison de la valeur de la variable \$a plus la valeur 3. (c'est la valeur qui est assignée à la variable \$a). N'importe quel opérateur binaire peut utiliser ce type d'assignation, par exemple '\$a -= 5' (soustraction de 5 de la valeur de la variable \$a), '\$b *= 7' (multiplication de la valeur de la variable \$b par 7).

Il y a une autre expression qui peut paraître complexe si vous ne l'avez pas vue dans d'autres langages, l'opérateur conditionnel ternaire :

L'opérateur conditionnel ternaire

```
<?php
    $first ? $second : $third
?>
```

Si la valeur de la première sous-expression est vraie (différente de 0), alors la deuxième sous-expression est évaluée et constitue le résultat de l'expression conditionnelle. Sinon, c'est la troisième sous-expression qui est évaluée et qui constitue le résultat de l'expression.

Les exemples suivants devraient vous permettre de mieux comprendre la pré-incrémentation, la post-incrémentation et le concept des expressions en général :

Pré et post-incrémentation : exemples

```
<?php
function double($i) {
    return $i*2;
}

$b = $a = 5;          /* Assigne la valeur 5 aux variables $a et $b */
$c = $a++;           /* Post-incrémentation de la variable $a et assignation de
    la valeur à la variable $c */
$e = $d = ++$b;      /* Pre-incrémentation, et assignation de la valeur aux
    variables $d et $e */
/* à ce niveau, les variables $d et $e sont égales à 6 */
$f = double($d++);   /* assignation du double de la valeur de $d à la variable $f ($f vaut 12),
    puis incrémentation de la valeur de $d */
$g = double(++$e);   /* assigne deux fois la valeur de $e après
    incrémentation, 2*7 = 14 to $g */
$h = $g += 10;       /* Tout d'abord, $g est incrémentée de 10, et donc $g vaut 24.
    Ensuite, la valeur de $g, (24) est assignée à la variable $h,
    qui vaut donc elle aussi 24. */
?>
```

Au début de ce chapitre, nous avons dit que nous allons décrire les différents types d'instructions, et donc, comme promis, nous allons voir que les expressions peuvent être des instructions. Mais, attention, toutes les expressions ne sont pas des instructions. Dans ce cas-là, une instruction est de la forme 'expr ;', c'est-à-dire, une expression suivie par un point-virgule. L'expression '\$b = \$a = 5;', '\$a = 5' est valide, mais ce n'est pas une instruction en elle-même. '\$b = \$a = 5' est une instruction valide.

La dernière chose qui mérite d'être mentionnée est la véritable valeur des expressions. Lorsque vous faites des tests sur une variable, dans une boucle conditionnelle par exemple, cela ne vous intéresse pas de savoir quelle est la valeur exacte de l'expression. Mais vous voulez seulement savoir si le résultat signifie TRUE ou FALSE (PHP n'a pas de type booléen). La véritable valeur d'une expression en PHP est calculée de la même manière qu'en Perl. Toute valeur numérique différente de 0 est considérée comme étant TRUE . Une chaîne de caractères vide et la chaîne de caractère 0 sont considérées comme FALSE . Toutes les autres valeurs sont vraies. Avec les types de variables non-scalaires (les tableaux et les objets), s'ils ne contiennent aucun élément, renvoient FALSE , sinon, ils renvoient TRUE .

PHP propose une implémentation complète et détaillée des expressions. PHP documente toutes ses expressions dans le manuel que vous êtes en train de lire. Les exemples qui vont suivre devraient vous donner une bonne idée de ce qu'est une expression et comment construire vos propres expressions. Dans tout ce qui va suivre, nous écrivons `expr` pour indiquer toute expression PHP valide.

5.6 Les opérateurs

Un opérateur est quelque chose que vous alimentez avec une ou plusieurs valeurs (ou expression, dans le jargon de programmation) qui retourne une autre valeur (donc que la construction elle-même devient une expression). Donc, vous pouvez penser aux fonctions ou constructions qui retournent une valeur (comme `print`) comme opérateur et ceux qui retournent rien du tout (comme `echo`).

Il y a trois types d'opérateurs. Le premier, l'opérateur unaire, qui opère sur une seule valeur, par exemple `!` (l'opérateur de négation) ou `++` (l'opérateur d'incrément). Le second groupe, les opérateurs binaires ; ce groupe contient la plus part des opérateurs supportés par PHP qui sont listés ci-dessous dans la section "[La précedence des opérateurs](#)".

Le troisième groupe est le groupe des opérateurs de terminaison : `?:`. Ils doivent être utilisés pour choisir entre deux expressions dépendants d'une troisième, plutôt que sélectionner deux phrases ou chemins d'exécution. Les expressions ternaires environnantes avec des parenthèses est une idée très bonne.

5.6.1 La précedence des opérateurs

La priorité des opérateurs spécifie l'ordre dans lequel les valeurs doivent être analysées. Par exemple, dans l'expression `1 + 5 * 3`, le résultat est 16 et non 18, car la multiplication ("`*`") a une priorité supérieure par rapport à l'addition ("`+`"). Des parenthèses peuvent être utilisées pour forcer la priorité, si nécessaire. Par exemple : `(1 + 5) * 3` donnera 18. Si la priorité d'opérateur est égale, l'associativité de gauche à droite est utilisée.

Le tableau suivant dresse une liste de la priorité des différents opérateurs dans un ordre croissant de priorité. Les opérateurs sur une même ligne ont une priorité équivalente et dans ce cas, leur association décide de l'ordre de leur évaluation.

Associativité	Opérateurs	Information additionnelle
non-associative	<code>new</code>	new
gauche	<code>[</code>	array
non-associatif	<code>++ --</code>	incrément/décrément
non-associatif	<code>! ~ - (int) (float) (string) (array) (object) @</code>	types
gauche	<code>* / %</code>	arithmétique
gauche	<code>+ - .</code>	arithmétique et chaîne de caractères
gauche	<code><< >></code>	bitwise
non-associatif	<code>< <= > >=</code>	comparaison
non-associatif	<code>== != === !==</code>	comparaison
gauche	<code>&</code>	bitwise et références
gauche	<code>^</code>	bitwise
gauche	<code> </code>	bitwise

gauche	&&	logique
gauche		logique
gauche	? :	ternaire
droite	= += -= *= /= .= %= &= = ^= <<= >>=	assignement
gauche	and	logique
gauche	xor	logique
gauche	or	logique
gauche	,	plusieurs utilisations

L'associativité de gauche signifie que l'expression est évaluée de gauche à droite, l'associativité de droite, l'inverse.

<p>Associativité</p> <pre><?php \$a = 3 * 3 % 5; // (3 * 3) % 5 = 4 \$a = true ? 0 : true ? 1 : 2; // (true ? 0 : true) ? 1 : 2 = 2 \$a = 1; \$b = 2; \$a = \$b += 3; // \$a = (\$b += 3) -> \$a = 5, \$b = 5 ?></pre>

Utilisez les parenthèses pour augmenter la lisibilité du code.

<p>Note</p> <p>Bien que ! soit prioritaire sur = , PHP va tout de même exécuter des expressions comme : if (!\$a = foo()) . Dans cette situation, le résultat de foo() sera placé dans la variable \$a .</p>

5.6.2 Les opérateurs arithmétiques

Vous rappelez-vous des opérations élémentaires apprises à l'école ? Ils fonctionnent comme cela.

Exemple	Nom	Résultat
-\$a	Negation	Opposé de \$a.
\$a + \$b	Addition	Somme de \$a et \$b .
\$a - \$b	Soustraction	Différence de \$a et \$b .
\$a * \$b	Multiplication	Produit de \$a et \$b .
\$a / \$b	Division	Quotient de \$a et \$b .
\$a % \$b	Modulo	Reste de \$a divisé par \$b .

L'opérateur de division ("/") retourne une valeur entière (le résultat d'une division entière) si les deux opérandes sont entiers (ou bien des chaînes converties en entier).

<p>Note</p> <p>Souvenez-vous que \$a % \$b est négatif si \$a est négatif.</p> <p>Voir aussi le manuel sur les fonctions mathématiques .</p>

5.6.3 Les opérateurs d'assignation

L'opérateur d'assignation le plus simple est le signe "=". Le premier réflexe est de penser que ce signe veut dire "égal à". Ce n'est pas le cas. Il signifie que l'opérande de gauche se voit affecter la valeur de l'expression qui est à droite du signe égal.

La valeur d'une expression d'assignation est la valeur assignée. Par exemple, la valeur de l'expression ' \$a = 3 ' est la valeur 3. Cela permet d'utiliser des astuces telles que :

Astuce d'assignation
<pre><?php \$a = (\$b = 4) + 5; // \$a est maintenant égal à 9, et \$b vaut 4. ?></pre>

En plus du simple opérateur d'assignation, il existe des "opérateurs combinés" pour tous les opérateurs arithmétiques et pour les opérateurs sur les chaînes de caractères. Cela permet d'utiliser la valeur d'une variable dans une expression et d'affecter le résultat de cette expression à cette variable. Par exemple :

Exemples avec les opérateurs d'assignation
<pre><?php \$a = 3; \$a += 5; // affecte la valeur 8 à la variable \$a correspond à l'instruction '\$a = \$a + 5'; \$b = "Bonjour "; \$b .= " tout le monde!"; // affecte la valeur "Bonjour tout le monde!" à // la variable \$b // identique à \$b = \$b." tout le monde!"; ?></pre>

On peut noter que l'assignation copie le contenu de la variable originale dans la nouvelle variable (assignation par valeur), ce qui fait que les changements de valeur d'une variable ne modifieront pas la valeur de l'autre. Cela peut se révéler important lors de la copie d'un grand tableau durant une boucle. Depuis PHP 4, l'assignation par référence est supporté, en utilisant la syntaxe \$var = &\$othervar; , mais ce n'était pas possible en PHP 3. L'assignation par référence signifie que les deux variables contiennent les mêmes données, et que la modification de l'une affecte l'autre et rien n'est copié nul part. Pour plus d'informations sur les références, lisez [l'explication sur les références](#)

5.6.4 Opérateurs sur les bits

Les opérateurs sur les bits vous permettent de manipuler les bits dans un entier. Si les paramètres de gauche et de droite sont des chaînes de caractères, l'opérateur de bits agira sur les valeurs ASCII de ces caractères.

Exemples avec les opérateurs sur les bits
<pre><?php echo 12 ^ 9; // Affiche '5' echo "12" ^ "9"; // Affiche le caractère d'effacement (ascii 8) // ('1' (ascii 49)) ^ ('9' (ascii 57)) = #8 echo "hallo" ^ "hello"; // Affiche la valeur asciiie : s #0 #4 #0 #0 #0 // 'a' ^ 'e' = #4 ?></pre>

Exemple	Nom	Résultat
\$a & \$b	ET (And)	Les bits positionnés à 1 dans \$a ET dans \$b sont positionnés à 1.
\$a \$b	OU (Or)	Les bits positionnés à 1 dans \$a OU \$b sont positionnés à 1.
\$a ^ \$b	Xor	Les bits positionnés à 1 dans \$a OU dans \$b mais pas dans les deux sont positionnés à 1.

<code>~ \$a</code>	NON (Not)	Les bits qui sont positionnés à 1 dans \$a sont positionnés à 0, et vice versa.
<code>\$a << \$b</code>	Décalage à gauche	Décale les bits de \$a , \$b fois sur la gauche (chaque décalage équivaut à une multiplication par 2).
<code>\$a >> \$b</code>	Décalage à droite	Décalage des bits de \$a , \$b fois par la droite (chaque décalage équivaut à une division par 2).
Attention		
N'effectuez pas de décalage à droite de plus de 32 bits sur les systèmes 32 bits. N'effectuez pas de décalage à droite dans le cas où le résultat est un nombre plus long que 32 bits.		

5.6.5 Opérateurs de comparaison

Les opérateurs de comparaison, comme leur nom l'indique, vous permettent de comparer deux valeurs. Vous devriez également être intéressé par les [tables de comparaisons de types](#) car ils montrent des exemples de beaucoup de types de comparaisons.

Exemple	Nom	Résultat
<code>\$a == \$b</code>	Egal	TRUE si \$a est égal à \$b .
<code>\$a === \$b</code>	Identique	TRUE si \$a est égal à \$b et qu'ils sont de même type (introduit en PHP 4).
<code>\$a != \$b</code>	Différent	TRUE si \$a est différent de \$b .
<code>\$a <> \$b</code>	Différent	TRUE si \$a est différent de \$b .
<code>\$a !== \$b</code>	Différent	TRUE si \$a est différent de \$b ou bien qu'ils ne sont pas du même type. (introduit en PHP 4)
<code>\$a < \$b</code>	Plus petit que	TRUE si \$a est strictement plus petit que \$b .
<code>\$a > \$b</code>	Plus grand	TRUE si \$a est strictement plus grand que \$b .
<code>\$a <= \$b</code>	Inférieur ou égal	TRUE si \$a est plus petit ou égal à \$b .
<code>\$a >= \$b</code>	Supérieur ou égal	TRUE si \$a est plus grand ou égal à \$b .

Si vous comparez un entier avec une chaîne, la chaîne est convertie en un nombre . Si vous comparez deux chaînes numériques, elles seront comparées en tant qu'entiers. Ces règles s'appliquent aussi à l'instruction [switch](#) .

```
<?php
var_dump(0 == "a"); // 0 == 0 -> true
var_dump("1" == "01"); // 1 == 1 -> true

switch ("a") {
case 0:
    echo "0";
    break;
case "a": // jamais évalué parce que "a" est déjà trouvé avec 0
    echo "a";
    break;
}
?>
```

Pour les différents types, la comparaison est faite en suivant la table suivante (dans l'ordre).

Type de l'opérande 1	Type de l'opérande 2	Résultat
	string	Convertit NULL en "", comparaison numérique ou lexicale

null ou chaîne de caractères		
booléen ou null	N'importe quoi	Convertit en booléen , FALSE < TRUE
objet	objet	Les classes internes peuvent définir leur propre comparaison, différentes classes ne sont pas comparables, les mêmes classes - la comparaison des propriétés se fait de la même façon que pour les tableaux (PHP 4), PHP 5 a son propre <u>comportement</u>
chaîne de caractères , ressource ou nombre	chaîne de caractères , ressource ou nombre	Transforme les chaînes de caractères et les ressources en nombres Translate strings and resources to numbers, maths habituelles
tableaux	tableaux	Le tableau avec le moins de membres est plus petit, si la clé de l'opérande 1 n'est pas trouvée dans l'opérande 2, alors les tableaux ne sont pas comparables, sinon - comparaison valeur par valeur (voir l'exemple suivant)
tableau	N'importe quoi	Le tableau est toujours plus grand
objet	N'importe quoi	L' objet est toujours plus grand

Transcription des comparaisons standards des tableaux

```

<?php
// Les tableaux sont comparés comme ceci avec les opérateurs standards de comparaison
function standard_array_compare($op1, $op2)
{
    if (count($op1) < count($op2)) {
        return -1; // $op1 < $op2
    } elseif (count($op1) > count($op2)) {
        return 1; // $op1 > $op2
    }
    foreach ($op1 as $key => $val) {
        if (!array_key_exists($key, $op2)) {
            return null; // incomparable
        } elseif ($val < $op2[$key]) {
            return -1;
        } elseif ($val > $op2[$key]) {
            return 1;
        }
    }
    return 0; // $op1 == $op2
}
?>

```

Voir aussi [strcasecmp](#) , [strcmp](#) les opérateurs de tableaux , et le chapitre sur les [types](#) .

5.6.5.1 L'opérateur ternaire

Un autre opérateur conditionnel est l'opérateur ternaire (":?").

Assignement d'une valeur par défaut

```

<?php
// Exemple d'utilisation pour l'opérateur ternaire
$action = (empty($_POST['action'])) ? 'défaut' : $_POST['action'];

// La ligne ci-dessus est identique à la condition suivante :
if (empty($_POST['action'])) {
    $action = 'défaut';
} else {

```

```

    $action = $_POST['action'];
}
?>

```

L'expression (expr1) ? (expr2) : (expr3) est évaluée à expr2 si expr1 est évalué à TRUE , et expr3 si expr1 est évalué à FALSE .

Note

Notez que l'opérateur ternaire est une instruction, et il n'est pas évalué en tant que variable, mais en tant que résultat de l'instruction. Il est important de savoir si vous voulez retourner une variable par référence. L'instruction `return $var == 42 ? $a : $b;` dans une fonction retournée par référencene fonctionnera donc pas et une alerte est émise dans les versions supérieures de PHP.

5.6.6 Opérateur de contrôle d'erreur

PHP supporte un opérateur de contrôle d'erreur : c'est @. Lorsque cet opérateur est ajouté en préfixe d'une expression PHP, les messages d'erreur qui peuvent être générés par cette expression seront ignorés.

Si l'option `track_errors` est activée, les messages d'erreurs générés par une expression seront sauvés dans la variable globale `$php_errormsg` . Cette variable sera écrasée à chaque erreur. Il faut alors la surveiller souvent pour pouvoir l'utiliser.

Opérateur de contrôle d'erreur

```

<?php
/* Erreur intentionnelle (le fichier n'existe pas): */
$mon_fichier = @file ('non_persistant_file') or
    die ("Impossible d'ouvrir le fichier : L'erreur est : '$php_errormsg');

// Cela fonctionne avec n'importe quelle expression, pas seulement les fonctions
$value = @$cache[$key];
// la ligne ci-dessus n'affichera pas d'alerte si la clé $key du tableau n'existe pas

?>

```

Note

L'opérateur @ ne fonctionne qu'avec les expressions. La règle générale de fonctionnement est la suivante : si vous pouvez prendre la valeur de quelque chose, vous pouvez le préfixer avec @. Par exemple, vous pouvez ajouter @ aux variables, fonctions, à `include` , aux constantes, etc. Vous ne pourrez pas le faire avec des éléments de langage tels que les classes, if et foreach , etc.

Voir aussi [error_reporting](#) et la section sur la [gestion d'erreurs](#) .

Attention

En fait, l'opérateur "@" va aussi désactiver les rapports d'erreurs critiques, qui stoppent l'exécution du script. Entre autres, si vous utilisez "@" pour supprimer les erreurs de certaines fonctions, et que cette fonction n'existe pas, ou qu'elle a été mal orthographiée, vous n'aurez aucune indication.

5.6.7 Opérateur d'exécution

PHP supporte un opérateur d'exécution : guillemets obliques ("``"). Notez bien qu'il ne s'agit pas de guillemets simples. PHP essaiera d'exécuter le contenu de ces guillemets obliques comme une commande shell. Le résultat sera retourné (i.e. : il ne sera pas simplement envoyé à la sortie standard, il peut être assigné à une variable). Utilisez les guillemets obliques revient à utiliser la fonction `shell_exec` .

Opérateur d'exécution
<pre><?php \$output = `ls -al`; echo "<pre>\$output</pre>"; ?></pre>
Note
Cet opérateur est désactivé lorsque le <u>safe mode</u> est activé ou bien que la fonction <u>shell_exec</u> est désactivée.

Voir aussi le manuel à la section sur les fonctions d'exécution système , popen , proc_open et l'utilisation de PHP en ligne de commande .

5.6.8 Opérateurs d'incrémentatation et décrémentation

PHP supporte les opérateurs de pre- et post-incrémentatation et décrémentation, comme en langage C.

Note		
Les opérateurs d'incrémentatation/décrémentatation n'affecte pas les valeurs booléennes. La décrémentation des valeurs NULL n'a également aucun effet, mais leur incrémentatation donnera comme résultat 1 .		
Exemple	Nom	Résultat
++\$a	Pre-incrémente	Incrémente \$a de 1, puis retourne \$a .
\$a++	Post-incrémente	Retourne \$a , puis l'incrémente de 1.
--\$a	Pré-décrémente	Décrémente \$a de 1, puis retourne \$a .
\$a--	Post-décrémente	Retourne \$a , puis décrémente \$a de 1.

Voici un exemple simple

Exemples d'incrémentations et décréments
<pre><?php echo '<h3>Post-incrémentatation</h3>'; \$a = 5; echo "Devrait valoir 5: " . \$a++ . "
\n"; echo "Devrait valoir 6: " . \$a . "
\n"; echo '<h3>Pre-incrémentatation</h3>'; \$a = 5; echo "Devrait valoir 6: " . ++\$a . "
\n"; echo "Devrait valoir 6: " . \$a . "
\n"; echo '<h3>Post-décémentatation</h3>'; \$a = 5; echo "Devrait valoir 5: " . \$a-- . "
\n"; echo "Devrait valoir 4: " . \$a . "
\n"; echo '<h3>Pre-décémentatation</h3>'; \$a = 5; echo "Devrait valoir 4: " . --\$a . "
\n"; echo "Devrait valoir 4: " . \$a . "
\n"; ?></pre>

PHP suit les conventions de Perl pour la gestion des opérateurs arithmétiques, et non pas celle du C. Par exemple, en Perl 'Z'+1 retourne 'AA' , alors qu'en C, 'Z'+1 retourne '[' (ord('Z') == 90 , donc ord '[') == 91). Notez que les variables de caractères peuvent être incrémentées, mais pas décrémentées.

Opérations arithmétiques sur un caractère

```
<?php
$i = 'W';
for($n=0; $n<6; $n++) {
    echo ++$i . "\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
X
Y
Z
AA
AB
AC
```

L'incrémentation ou la décrémentation d'un booléen n'a aucun effet.

5.6.9 Les opérateurs logiques

Exemple	Nom	Résultat
<code>\$a and \$b</code>	ET (And)	Vrai si \$a ET \$b sont vrais.
<code>\$a or \$b</code>	OU (Or)	Vrai si \$a OU \$b est vrai
<code>\$a xor \$b</code>	XOR (Xor)	Vrai si \$a OU \$b est vrai, mais pas les deux en même temps.
<code>! \$a</code>	NON (Not)	Vrai si \$a est faux.
<code>\$a && \$b</code>	ET (And)	Vrai si \$a ET \$b sont vrais.
<code>\$a \$b</code>	OU (Or)	Vrai si \$a OU \$b est vrai.

La raison pour laquelle il existe deux types de "ET" et de "OU" est qu'ils ont des priorités différentes. Voir le paragraphe [précédence d'opérateurs](#) .

5.6.10 Opérateurs de chaînes

Il y a deux opérateurs de chaînes de caractères string . Le premier est l'opérateur de concaténation (.), qui retourne la concaténation de ses deux arguments. Le second est l'opérateur d'assignation concaténant (.=). Reportez-vous à [opérateurs d'assignation](#) pour plus de détails.

Opérateur de concaténation

```
<?php
$a = "Bonjour ";
$b = $a . "Monde!"; // $b contient "Bonjour Monde!"

$a = "Bonjour ";
$a = $a . "Monde!"; // $a contient "Bonjour Monde!"
?>
```

Voir aussi les sections du manuel sur [les types de chaînes de caractères](#) et [les chaînes de caractères](#) .

5.6.11 Opérateurs de tableaux

Exemple	Nom	Résultat
<code>\$a + \$b</code>	Union	Union de \$a et \$b .
<code>\$a == \$b</code>	Egalité	TRUE si \$a et \$b contiennent les mêmes paires clés / valeurs.
<code>\$a === \$b</code>	Identique	TRUE si \$a et \$b contiennent les mêmes paires clés / valeurs dans le même ordre et du même type.
<code>\$a != \$b</code>	Inégalité	TRUE si \$a n'est pas égal à \$b .
<code>\$a <> \$b</code>	Inégalité	TRUE si \$a n'est pas égal à \$b .
<code>\$a !== \$b</code>	Non-identique	TRUE si \$a n'est pas identique à \$b .

Le seul opérateur en PHP pour les tableaux est l'opérateur `+` . Cet opérateur ajoute le tableau de droite au tableau de gauche. Les clés communes sont laissées inchangées.

Addition de tableaux
<pre><?php \$a = array("a" => "pomme", "b" => "banane"); \$b = array("a" => "poire", "b" => "fraise", "c" => "cerise"); \$c = \$a + \$b; // Union de \$a et \$b echo "Union de \\$a et \\$b : \n"; var_dump(\$c); \$c = \$b + \$a; // Union de \$b et \$a echo "Union de \\$b et \\$a : \n"; var_dump(\$c); ?></pre>

L'exemple ci-dessus va afficher :

Union de \$a et \$b :

```
array(3) {
  ["a"]=>
  string(5) "pomme"
  ["b"]=>
  string(6) "banane"
  ["c"]=>
  string(6) "cerise"
}
```

Union de \$b et \$a :

```
array(3) {
  ["a"]=>
  string(5) "poire"
  ["b"]=>
  string(6) "fraise"
  ["c"]=>
  string(6) "cerise"
}
```

Les éléments d'un tableau sont égaux en terme de comparaison s'ils ont la même clé et la même valeur.

Comparer des tableaux
<pre><?php \$a = array("pomme", "banane");</pre>

```

$b = array(1 => "banane", "0" => "pomme");

var_dump($a == $b); // bool(true)
var_dump($a === $b); // bool(false)
?>

```

Voyez aussi le manuel aux sections [Tableaux](#) et [fonctions de tableaux](#) .

5.6.12 Opérateur de type

PHP a un seul opérateur de type : `instanceof` est utilisé pour déterminer si oui ou non un objet donné, son parent ou ses [interfaces](#) implémentées sont d'une [classe objet](#) spécifique.

L'opérateur `instanceof` a été introduit en PHP 5. Avant cela, `is_a` été utilisé mais il est désormais déconseillé de l'utiliser, en faveur de `instanceof` .

Opérateur de type : instanceof

```

<?php
class A { }
class B { }

$thing = new A;

if ($thing instanceof A) {
    echo 'A';
}
if ($thing instanceof B) {
    echo 'B';
}
?>

```

Comme `$thing` est un [objet](#) de type A, mais pas B, seul le premier bloc de commande sera exécuté.

A

Voir aussi [get_class](#) et [is_a](#) .

5.7 Les structures de contrôle

Tous les scripts PHP sont une suite d'instructions. Une instruction peut être une assignation, un appel de fonction, une instruction conditionnelle ou bien une instruction qui ne fait rien (une instruction vide). Une instruction se termine habituellement par un point virgule (" ; "). De plus, plusieurs instructions peuvent être regroupées en bloc, délimité par des accolades (" {} "). Un bloc est considéré comme une instruction. Les différents types d'instruction sont décrits dans ce chapitre.

5.7.1 if

L'instruction `if` est une des plus importantes instructions de tous les langages, PHP inclus. Elle permet l'exécution conditionnelle d'une partie de code. Les fonctionnalités de l'instruction `if` sont les mêmes en PHP qu'en C :

Instruction if ()

```
if (expression)
    commandes
```

Comme nous l'avons vu dans le paragraphe consacré aux [expressions](#) , expression est convertie en sa valeur booléenne. Si l' expression vaut TRUE , PHP exécutera l' instruction et si elle vaut FALSE , l'instruction sera ignorée. Plus de détails sur les valeurs qui valent FALSE sont disponibles dans la section [Conversion en booléen](#) .

L'exemple suivant affiche la phrase a est plus grand que b si \$a est plus grand que \$b :

Instruction if () (2)

```
<?php
if ($a > $b)
    print "a est plus grand que b";
?>
```

Souvent, vous voulez que plusieurs instructions soient exécutées après un branchement conditionnel. Bien évidemment, il n'est pas obligatoire de répéter l'instruction conditionnelle autant de fois que vous avez d'instructions à exécuter. A la place, vous pouvez rassembler toutes les instructions dans un bloc. L'exemple suivant affiche a est plus grand que b , et assigne la valeur de la variable \$a à la variable \$b :

Instruction if () et bloc

```
<?php
if ($a > $b) {
    echo "a est plus grand que b";
    $b = $a;
}
?>
```

Vous pouvez imbriquer indéfiniment des instructions if les unes dans les autres, ce qui permet une grande flexibilité dans l'exécution d'une partie de code suivant un grand nombre de conditions.

5.7.2 else

Souvent, vous voulez exécuter une instruction si une condition est remplie, et une autre instruction si cette condition n'est pas remplie. C'est à cela que sert else . else fonctionne après un if et exécute les instructions correspondantes au cas où l'expression du if est FALSE . Dans l'exemple suivant, ce bout de code affiche a est plus grand que b si la variable \$a est plus grande que la variable \$a , et a est plus petit que b sinon :

Instruction if () else

```
<?php
if ($a > $b) {
    echo "a est plus grand que b";
} else {
    echo "a est plus petit que b";
}
?>
```

Les instructions après le else ne sont exécutées que si l'expression du if est FALSE , et si elle n'est pas suivi par l'expression elseif - uniquement si elles sont évaluées à FALSE (voir [elseif](#)).

5.7.3 elseif

elseif , comme son nom l'indique, est une combinaison de if et de else . Comme l'expression else , il permet d'exécuter une instruction après un if dans le cas où le "premier" if est évalué comme FALSE . Mais, à la différence de l'expression else , il n'exécutera l'instruction que si l'expression conditionnelle elseif est évaluée comme TRUE . L'exemple suivant affichera a est plus grand que b , a est égal à b ou a est plus petit que b :

Instruction elseif
<pre><?php if (\$a > \$b) { echo "a est plus grand que b"; } elseif (\$a == \$b) { echo "a est égal à b"; } else { echo "a est plus petit que b"; } ?></pre>

Vous pouvez avoir plusieurs elseif qui se suivent les uns après les autres, après un if initial. Le premier elseif qui sera évalué à TRUE sera exécuté. En PHP, vous pouvez aussi écrire " else if " en deux mots et son comportement sera identique à la version en un seul mot. La sémantique des deux expressions est légèrement différente, mais au bout du compte, le résultat sera exactement le même.

L'expression elseif est exécutée seulement si le if précédent et tout autre elseif précédent sont évalués comme FALSE , et que votre elseif est évalué à TRUE .

5.7.4 Syntaxe alternative

PHP propose une autre manière de rassembler des instructions à l'intérieur d'un bloc, pour les fonctions de contrôle if , while , for , foreach et switch . Dans chaque cas, le principe est de remplacer l'accolade d'ouverture par deux points (:) et l'accolade de fermeture par, respectivement, endif; , endwhile; , endfor; ou endswitch; .

Instruction alternative if()
<pre><?php if (\$a == 5): ?> A égal 5 <?php endif; ?></pre>

Dans l'exemple ci-dessus, le bloc HTML "A égal 5" est inclus à l'intérieur d'un if en utilisant cette nouvelle syntaxe. Ce code HTML ne sera affiché que si la variable \$a est égale à 5.

Cette autre syntaxe fonctionne aussi avec le else et elseif . L'exemple suivant montre une structure avec un if , un elseif et un else utilisant cette autre syntaxe :

Instruction alternative if() elseif() et else
<pre><?php if (\$a == 5): echo "a égale 5"; echo "..."; elseif (\$a == 6): echo "a égale 6"; echo "!!!"; else: echo "a ne vaut ni 5 ni 6";</pre>

```
endif;
?>
```

Voir aussi [while](#) , [for](#) , et [if](#) pour d'autres exemples.

5.7.5 while

La boucle while est le moyen le plus simple d'implémenter une boucle en PHP. Cette boucle se comporte de la même manière qu'en C. L'exemple le plus simple d'une boucle while est le suivant :

Instruction while

```
while (expression)
    commandes
```

La signification d'une boucle while est très simple. PHP exécute l'instruction tant que l'expression de la boucle while est évaluée comme TRUE . La valeur de l'expression est vérifiée à chaque début de boucle, et, si la valeur change durant l'exécution de l'instruction, l'exécution ne s'arrêtera qu'à la fin de l'itération (chaque fois que PHP exécute l'instruction, on appelle cela une itération). De temps en temps, si l'expression du while est FALSE avant la première itération, l'instruction ne sera jamais exécutée.

Comme avec le if , vous pouvez regrouper plusieurs instructions dans la même boucle while en les regroupant à l'intérieur de parenthèses ou en utilisant la syntaxe suivante :

Instruction while avec bloc

```
while (expression):
    commandes
    ...
endwhile;
```

Les exemples suivants sont identiques et affichent tous les nombres de 1 à 10 :

Exemple avec while

```
<?php
/* exemple 1 */

$i = 1;
while ($i <= 10) {
    echo $i++; /* La valeur affiche est $i avant l'incrémentaion
                (post-incrémentaion) */
}

/* exemple 2 */

$i = 1;
while ($i <= 10):
    echo $i;
    $i++;
endwhile;
?>
```

5.7.6 do-while

Les boucles do-while ressemblent beaucoup aux boucles while , mais l'expression est testée à la fin de chaque itération plutôt qu'au début. La principale différence par rapport à la boucle while est que la première itération de la boucle do-while est toujours exécutée (l'expression n'est testée qu'à la fin de l'itération), ce qui n'est pas le cas lorsque vous utilisez une boucle while (l'expression est vérifiée

au début de chaque itération).

Il n'y a qu'une syntaxe possible pour les boucles do-while :

Instruction do-while

```
<?php
$i = 0;
do {
    echo $i;
} while ($i > 0);
?>
```

La boucle ci-dessus ne va être exécutée qu'une seule fois, car lorsque l'expression est évaluée, elle vaut FALSE (car la variable \$i n'est pas plus grande que 0) et l'exécution de la boucle s'arrête.

Les utilisateurs familiers du C sont habitués à une utilisation différente des boucles do-while , qui permet de stopper l'exécution de la boucle au milieu des instructions, en l'encapsulant dans un do-while (0) la fonction `break` . Le code suivant montre une utilisation possible :

Instruction while et break

```
<?php
do {
    if ($i < 5) {
        echo "i n'est pas suffisamment grand";
        break;
    }
    $i *= $factor;
    if ($i < $minimum_limit) {
        break;
    }
    echo "i est bon";

    /* ...traitement de i... */
} while (0);
?>
```

Ne vous inquiétez pas si vous ne comprenez pas tout correctement. Vous pouvez écrire des scripts très très puissants sans utiliser cette fonctionnalité.

5.7.7 for

Les boucles for sont les boucles les plus complexes en PHP. Elles fonctionnent comme les boucles for du langage C. La syntaxe des boucles for est la suivante :

Instruction for

```
for (expr1; expr2; expr3)
    commandes
```

La première expression (`expr1`) est évaluée (exécutée), quoi qu'il arrive au début de la boucle.

Au début de chaque itération, l'expression `expr2` est évaluée. Si l'évaluation vaut TRUE , la boucle continue et l'instruction est exécutée. Si l'évaluation vaut FALSE , l'exécution de la boucle s'arrête.

A la fin de chaque itération, l'expression `expr3` est évaluée (exécutée).

Les expressions peuvent éventuellement être laissées vides. Si l'expression `expr2` est laissée vide, cela signifie que c'est une boucle infinie (PHP considère implicitement qu'elle vaut TRUE , comme

en C). Cela n'est pas vraiment très utile, à moins que vous souhaitiez terminer votre boucle par l'instruction conditionnelle `break`.

Considérons les exemples suivants. Tous affichent les chiffres de 1 à 10 :

Exemple de boucles
<pre> <?php /* exemple 1 */ for (\$i = 1; \$i <= 10; \$i++) { echo \$i; } /* exemple 2 */ for (\$i = 1; ; \$i++) { if (\$i > 10) { break; } echo \$i; } /* exemple 3 */ \$i = 1; for (; ;) { if (\$i > 10) { break; } echo \$i; \$i++; } /* exemple 4 */ for (\$i = 1; \$i <= 10; print \$i, \$i++); ?> </pre>

Bien évidemment, le premier exemple est le plus simple de tous (ou peut être le quatrième), mais vous pouvez aussi penser qu'utiliser une expression vide dans une boucle for peut être utile parfois.

PHP supporte aussi la syntaxe alternative suivante pour les boucles for :

Instruction for alternative
<pre> for (expr1; expr2; expr3): commandes; ...; endfor; </pre>

5.7.8 foreach

PHP 4 introduit une commande foreach, comme en Perl ou d'autres langages. C'est un moyen simple de passer en revue un tableau. Il y a deux syntaxes possibles : la seconde est une extension mineure mais pratique de la première.

Instruction foreach
<pre> foreach(array_expression as \$value) commandes foreach(array_expression as \$key => \$value) </pre>

commandes

La première forme passe en revue le tableau `array_expression` . A chaque itération, la valeur de l'élément courant est assignée à `$value` et le pointeur interne de tableau est avancé d'un élément (ce qui fait qu'à la prochaine itération, on accédera à l'élément suivant).

La deuxième forme fait exactement la même chose, mais c'est la clé de l'élément courant qui est assigné à la variable `$key` .

Depuis PHP 5, il est possible d' itérer également des objets .

Note

Lorsque `foreach` démarre, le pointeur interne de fichier est automatiquement ramené au premier élément du tableau. Cela signifie que vous n'aurez pas à faire appel à reset avant `foreach` .

Note

Bien que le tableau soit référéncé , `foreach` travaille sur une copie du tableau spécifié, et pas sur le tableau lui-même. Par conséquent, le pointeur de tableau n'est pas modifié, comme il le serait avec la fonction each , et les modifications faites dans le tableau ne seront pas prises en compte dans le tableau original. Cependant, le pointeur interne du tableau original **est** avancé avec le traitement du tableau. En supposant sur la boucle `foreach` effectuée tout le traitement, le pointeur interne du tableau sera à la fin du tableau.

Depuis PHP 5, vous pouvez facilement modifier les éléments d'un tableau en faisant précéder la valeur `$value` avec `&`. Ceci aura pour effet d'assigner par référence au lieu de copier la valeur.

```
<?php
$arr = array(1, 2, 3, 4);
foreach ($arr as &$value) {
    $value = $value * 2;
}
// $arr vaut maintenant array(2, 4, 6, 8)
?>
```

Ceci n'est possible que si le tableau itéré peut être référéncé (i.e. est une variable).

Note

`foreach` n'accepte pas l'opérateur de suppression des erreurs `@` .

Vous pouvez remarquer que les exemples suivants fonctionnent de manière identique :

Instruction `foreach` et `while`

```
<?php
$arr = array("un", "deux", "trois");
reset($arr);
while (list(, $value) = each($arr)) {
    echo "Valeur : $value<br />\n";
}

foreach ($arr as $value) {
    echo "Valeur : $value<br />\n";
}
?>
```

Les exemples suivants sont aussi fonctionnellement identiques :

Instruction `foreach` et `while` (2)

```
<?php
$arr = array("un", "deux", "trois");
```

```

reset($arr);
while (list($key, $value) = each($arr)) {
    echo "Clé : $key; Valeur : $value<br />\n";
}

foreach ($arr as $key => $value) {
    echo "Clé : $key; Valeur : $value<br />\n";
}
?>

```

Voici quelques exemples de plus :

Instruction foreach et while (3)

```

<?php
/* exemple foreach 1 : la valeur seulement */

$a = array(1, 2, 3, 17);

foreach ($a as $v) {
    echo "Current value of \$a : $v.\n";
}

/* exemple foreach 2 : la valeur et sa clé d'index */

$a = array(1, 2, 3, 17);

$i = 0; /* uniquement pour l'illustration */

foreach ($a as $v) {
    echo "\$a[$i] => $v.\n";
    $i++;
}

/* exemple foreach 3 : la clé et la valeur */

$a = array (
    "un" => 1,
    "deux" => 2,
    "trois" => 3,
    "dix-sept" => 17
);

foreach ($a as $k => $v) {
    echo "\$a[$k] => $v.\n";
}

/* exemple foreach 4 : tableaux multi-dimensionnels */

$a = array();
$a[0][0] = "a";
$a[0][1] = "b";
$a[1][0] = "y";
$a[1][1] = "z";

foreach ($a as $v1) {
    foreach ($v1 as $v2) {
        echo "$v2\n";
    }
}

/* exemple foreach 5 : tableaux dynamiques */

foreach (array(1, 2, 3, 4, 5) as $v) {
    echo "$v\n";
}

```

```
?>
```

5.7.9 break

L'instruction `break` permet de sortir d'une structure `for` , `foreach` , `while` , `do-while` ou `switch` .

`break` accepte un argument numérique optionnel qui vous indiquera combien de structures emboîtées ont été interrompues.

Instruction break

```
<?php
$arr = array('un', 'deux', 'trois', 'quatre', 'stop', 'cinq');
while (list(, $val) = each($arr)) {
    if ($val == 'stop') {
        break;    /* Vous pourriez aussi utiliser 'break 1;' ici. */
    }
    echo "$val<br />\n";
}

/* Utilisation de l'argument optionnel. */

$i = 0;
while (++$i) {
    switch ($i) {
        case 5:
            echo "A 5<br />\n";
            break 1; /* Termine uniquement le switch. */
        case 10:
            echo "A 10; quitter<br />\n";
            break 2; /* Termine le switch et la boucle while. */
        default:
            break;
    }
}
?>
```

5.7.10 continue

L'instruction `continue` est utilisée dans une boucle afin d'éviter les instructions de l'itération courante et de continuer l'exécution à la condition de l'évaluation et donc, de commencer la prochaine itération.

Note

Notez qu'en PHP, la structure `switch` est considérée comme une boucle `continue` .

`continue` accepte un argument numérique optionnel qui vous indiquera combien de structures emboîtées ont été ignorées.

Instruction continue

```
<?php
while (list($cle, $valeur) = each($arr)) {
    if (!($cle % 2)) { // évite les membres impairs
        continue;
    }
    fonction_quelconque($valeur);
}

```

```

$i = 0;
while ($i++ < 5) {
    echo "Dehors<br />\n";
    while (1) {
        echo "&nbsp;&nbsp;&nbsp;Milieu<br />\n";
        while (1) {
            echo "&nbsp;&nbsp;&nbsp;Intérieur<br />\n";
            continue 3;
        }
        echo "Ceci n'est jamais atteint.<br />\n";
    }
    echo "Ceci non plus.<br />\n";
}
?>

```

Oublier le point virgule après continue peut porter à confusion. Voici un exemple de ce que vous ne devez pas faire :

N'oubliez pas le point-virgule après continue

```

<?php
for ($i = 0; $i < 5; ++$i) {
    if ($i == 2)
        continue
    print "$i\n";
}
?>

```

On peut s'attendre à ce que le résultat soit :

```

0
1
3
4

```

mais ce script affichera :

```

2

```

car la valeur de retour de l'appel à `print` est `int(1)` , et cela se comportera alors comme si on avait fourni l'argument optionnel mentionné plus haut.

5.7.11 switch

L'instruction `switch` équivaut à une série d'instructions `if` . En de nombreuses occasions, vous aurez besoin de comparer la même variable (ou expression) avec un grand nombre de valeurs différentes, et d'exécuter différentes parties de code suivant la valeur à laquelle elle est égale. C'est exactement à cela que sert l'instruction `switch` .

Note

Notez que contrairement à d'autres langages, la structure `continue` s'applique aux structures `switch` et de comporte différemment de `break` . Si vous avez un `switch` dans une boucle, et que vous souhaitez continuer jusqu'à la prochaine itération de la boucle extérieure, vous vous devez utiliser `continue 2` .

Les deux exemples suivants sont deux manières différentes d'écrire la même chose, l'une en utilisant une série de if , et l'autre en utilisant l'instruction switch :

Instruction switch

```
<?php
if ($i == 0) {
    echo "i égale 0";
} elseif ($i == 1) {
    echo "i égale 1";
} elseif ($i == 2) {
    echo "i égale 2";
}

switch ($i) {
case 0:
    echo "i égale 0";
    break;
case 1:
    echo "i égale 1";
    break;
case 2:
    echo "i égale 2";
    break;
}
?>
```

Instruction switch utilisant une chaîne de caractères

```
<?php
switch ($i) {
case "tarte":
    echo "i est une tarte";
    break;
case "barre":
    echo "i est une barre";
    break;
case "gateau":
    echo "i est un gateau";
    break;
}
?>
```

Il est important de comprendre que l'instruction switch exécute chacune des clauses dans l'ordre. L'instruction switch est exécutée ligne par ligne. Au début, aucun code n'est exécuté. Seulement lorsqu'un case est vérifié, PHP exécute alors les instructions correspondantes. PHP continue d'exécuter les instructions jusqu'à la fin du bloc d'instructions du switch , ou bien dès qu'il trouve l'instruction break . Si vous ne pouvez pas utiliser l'instruction break à la fin de l'instruction case , PHP continuera à exécuter toutes les instructions qui suivent. Par exemple :

Instruction switch et fall through

```
<?php
switch ($i) {
case 0 :
    echo "i égale 0";
case 1 :
    echo "i égale 1";
case 2 :
    echo "i égale 2";
}
?>
```

Dans cet exemple, si \$i est égal à 0, PHP va exécuter quand même toutes les instructions qui suivent! Si \$i est égal à 1, PHP exécutera les deux dernières instructions. Et seulement si \$i est

égal à 2, vous obtiendrez le résultat escompté, c'est-à-dire, l'affiche de "i égal 2". Donc, l'important est de ne pas oublier l'instruction break (même s'il est possible que vous l'omettiez dans certaines circonstances).

Dans une commande switch , une condition n'est évaluée qu'une fois, et le résultat est comparé à chaque case . Dans une structure elseif , les conditions sont évaluées à chaque comparaison. Si votre condition est plus compliquée qu'une simple comparaison, ou bien fait partie d'une boucle, switch sera plus rapide.

La liste de commandes d'un case peut être vide, auquel cas PHP utilisera la liste de commandes du cas suivant.

Instruction switch et cas vide

```
<?php
switch ($i) {
case 0:
case 1:
case 2:
    echo "i est plus petit que 3 mais n'est pas négatif";
    break;
case 3 :
    echo "i égale 3";
}
?>
```

Un case spécial est default . Ce cas est utilisé lorsque tous les case ont échoués. Il doit être le dernier cas listé. Par exemple :

Instruction switch et default

```
<?php
switch ($i) {
case 0:
    echo "i égale 0";
    break;
case 1:
    echo "i égale 1";
    break;
case 2:
    echo "i égale 2";
    break;
default:
    echo "i n'est ni égal à 2, ni à 1, ni à 0.";
}
?>
```

Une autre chose à mentionner est que la valeur du case peut être toute expression de type scalaire, c'est-à-dire nombre entier, nombre à virgule flottante et chaîne de caractères. Les tableaux sont sans intérêt dans ce contexte-là.

La syntaxe alternative pour cette structure de contrôle est la suivante : pour plus d'informations, voir [syntaxes alternatives](#)).

Instruction switch alternative

```
<?php
switch ($i):
case 0:
    echo "i égale 0";
    break;
case 1:
    echo "i égale 1";
```

```

    break;
case 2:
    echo "i égale 2";
    break;
default:
    echo "i n'est ni égal à 2, ni à 1, ni à 0";
endswitch;
?>

```

5.7.12 declare

L'élément de langage `declare` sert à ajouter des directives d'exécutions dans un bloc de code. La syntaxe de `declare` est similaire à la syntaxe des autres fonctions de contrôle :

Instruction directive

```
declare (directive)
    commandes
```

L'expression `directive` permet de contrôler l'intervention du bloc `declare` . Actuellement, une seule directive est reconnue : la directive `ticks` (Voir plus bas pour plus de détails) sur les [ticks](#)).

L'expression `commandes` du bloc de `declare` sera exécutée. Comment elle sera exécutée, et quels effets cela aura dépend de la directive utilisée dans le bloc directive .

La structure `declare` peut aussi être utilisée dans le contexte global. Elle affecte alors tout le code qui la suit :

Instruction directive en contexte global

```

<?php
// Ces déclaration sont identiques.

// Vous pouvez utiliser ceci
declare(ticks=1) {
    // script entier ici
}

// ou ceci
declare(ticks=1);
// script entier ici
?>

```

5.7.12.1 Ticks

Un tick est un événement qui intervient toutes les N commandes bas niveau, exécutées par l'analyseur dans le bloc de `declare` . La valeur de N est spécifiée avec la syntaxe `ticks= N` dans le bloc de directive `declare` .

Un événement qui intervient à chaque tick est spécifié avec la fonction [register_tick_function](#) . Reportez-vous à l'exemple ci-dessous pour plus de détails. Notez que plus d'un événement peut intervenir par tick.

Profiler une section de code PHP

```

<?php
// Un fonction qui enregistre l'heure à laquelle elle est appelée
function profile($dump = FALSE){
    static $profile;
    // Retourne les horaires stockés dans le profile, et l'efface

```

```

    if ($dump) {
        $temp = $profile;
        unset($profile);
        return $temp;
    }
    $profile[] = microtime ();
}

// Enregistre un gestionnaire de tick
register_tick_function("profile");

// Initialise la fonction avant le bloc de déclaration
profile();

// Exécute un bloc de code, et appelle un tick toutes les deux secondes
declare (ticks=2) {
    for ($x = 1; $x < 50; ++$x) {
        echo similar_text(md5($x), md5($x*$x)), "<br />";
    }
}

// Affiche les données de la variable $profile
print_r(profile(TRUE));
?>

```

Cet exemple profile le code PHP dans le bloc de déclaration, et enregistre l'heure de chaque commande bas niveau. Cette information peut être réutilisée pour débusquer les segments de code lents. Vous pouvez implémenter d'autres méthodes, mais les ticks sont plus rapides et plus efficaces.

Les ticks sont bien pratiques pour déboguer, implémenter un multi-tâches simple, des entrées sorties en tâche de fond, ou bien d'autres choses, avec PHP.

Voir aussi [register_tick_function](#) et [unregister_tick_function](#) .

5.7.13 return

Si appelée depuis une fonction, la commande [return](#) termine immédiatement la fonction, et retourne l'argument qui lui est passé. [return](#) interrompt aussi l'exécution de commande [eval](#) ou de scripts.

Si appelée depuis l'environnement global, l'exécution du script est interrompue. Si le script courant était [include](#) ou [require](#) , alors le contrôle est rendu au script appelant, et la valeur retournée sera utilisée comme résultat de la fonction [include](#) . Si [return](#) est appelée depuis le script principal, alors l'exécution du script s'arrête. Si le script courant est [auto_prepend_file](#) ou [auto_append_file](#) dans le fichier php.ini , alors l'exécution du script s'arrête.

Pour plus d'informations, voyez [retourner des valeurs](#) .

Note

Notez que puisque [return](#) est une structure de langage et non pas une fonction, les parenthèses **ne sont pas** requises : en fait, il est classique de les ignorer, même si cela ne change rien et vous devriez actuellement en faire de même, PHP ayant moins de travail dans ce cas.

Note

Vous ne devriez **jamais** utiliser les parenthèses autant de la variable retournée lorsque vous la retournez par référence, car cela ne fonctionnera pas. Vous pouvez uniquement retourner des variables par référence, et non pas le résultat du traitement. Si vous utilisez `return ($a);` , alors vous ne retournez pas une variable mais le résultat de l'expression `($a)` (qui est, bien sûr, la valeur de `$a`

).

5.7.14 require

`require` inclut et exécute un fichier PHP.

La commande `require` se remplace elle-même par le contenu du fichier spécifié, tel que décrit dans la documentation de la fonction `include` .

`require` et `include` sont identiques, sauf dans leur façon de gérer les erreurs. `include` produit une **Alerte (warning)** tandis que `require` génère une **erreur fatale** . En d'autres termes, n'hésitez pas à utiliser `require` si vous voulez qu'un fichier d'inclusion manquant interrompe votre script. `include` ne se comporte pas de cette façon, et le script continuera son exécution. Assurez-vous d'avoir bien configuré le `include_path` aussi.

Exemples avec `require`

```
<?php
require 'prepend.php';
require $un_fichier;
require ('un_fichier.txt');
?>
```

Voyez la documentation de la fonction `include` pour plus d'exemples.

Note

Avant PHP 4.0.2, ceci s'appliquait : `require` tentait de lire le fichier cible, même si les lignes n'étaient pas utilisées. Une condition `if` n'avait aucun effet sur `require` . Cependant, si la ligne du `require` n'est jamais utilisée, le code inclus ne le sera pas non plus. De même, les boucles n'ont pas d'effet sur le comportement de `require` . Bien que le code contenu dans le fichier subit la boucle, la fonction elle-même `require` n'est appelée qu'une fois.

Note

Comme ceci est une structure du langage, et non pas une fonction, il n'est pas possible de l'appeler avec les variables fonctions .

Attention

Les versions Windows de PHP antérieures à la version 4.3.0 ne supportent pas l'accès aux fichiers distants avec cette fonction, même si `allow_url_fopen` est activé.

Voir aussi `include` , `require_once` , `include_once` , `eval` , `file` , `readfile` , `virtual` et `include_path` .

5.7.15 include

La fonction `include` inclut et exécute le fichier spécifié en argument.

Cette documentation s'applique aussi à la fonction `require` . Les deux structures de langage sont identiques, hormis dans leur gestion des erreurs. `include` produit une **Alerte (warning)** tandis que `require` génère une **erreur fatale** . En d'autres termes, n'hésitez pas à utiliser `require` si vous voulez qu'un fichier d'inclusion manquant interrompe votre script. `include` ne se comporte pas de cette façon, et le script continuera son exécution. Assurez-vous d'avoir bien configuré le `include_path` aussi. Soyez prévenus qu'une erreur d'analyse dans un fichier inclut ne cause pas l'arrêt du script en PHP dans les versions antérieures à 4.3.5. Depuis ces versions, il le peut.

Les fichiers à inclure sont d'abord recherchés dans le dossier désigné par `include_path`, relativement au dossier courant, puis dans `include_path`, relativement au dossier de travail du script. Par exemple, si `include_path` est `.`, que le dossier de travail est `/www/`, et que vous incluez le fichier `include/a.php` et qu'il y a une instruction `include "b.php"` dans ce fichier, alors `b.php` est d'abord recherché dans `/www/`, puis dans `/www/include/`. Si le nom du fichier commence par `./` ou `../`, il est cherché uniquement dans l'`include_path` relatif au dossier courant d'exécution.

Lorsqu'un fichier est inclus, le code le composant hérite de la portée des variables de la ligne où l'inclusion apparaît. Toutes les variables disponibles à cette ligne dans le fichier appelant seront disponibles dans le fichier appelé, à partir de ce point. Cependant, toutes les fonctions et classes définies dans le fichier inclu ont une portée globale.

Exemple avec `include`

```
vars.php
<?php

$couleur = 'verte';
$fruit = 'pomme';

?>

test.php
<?php

echo "Une $couleur $fruit"; // Une

include 'vars.php';

echo "Une $couleur $fruit"; // Une verte pomme

?>
```

Si l'inclusion intervient à l'intérieure d'une fonction, le code inclus sera alors considéré comme faisant partie de la fonction. Cela modifie donc le contexte de variables accessibles.

Inclusion de fichiers dans une fonction

```
<?php

function foo() {
    global $couleur;

    include 'vars.php';

    echo "Une $couleur $fruit";
}

/* vars.php est dans le contexte de foo()    *
 * donc $fruit n'est pas disponibles hors de *
 * cette fonction. $couleur l'est, car c'est *
 * une variable globale                      */

foo(); // Une verte pomme
echo "Une $couleur $fruit"; // Une verte

?>
```

Il est important de noter que lorsqu'un fichier est `include` ou `require`, les erreurs d'analyse apparaîtront en HTML tout au début du fichier, et l'analyse du fichier parent ne sera pas interrompue. Pour cette raison, le code qui est dans le fichier doit être placé entre les balises habituelles de PHP.

Si les Gestionnaires d'URL sont activés dans PHP (ce qui est le cas par défaut), vous pouvez localiser le fichier avec une URL (via HTTP ou bien avec un gestionnaire adapté : voir Liste des protocoles supportés pour une liste des protocoles), au lieu d'un simple chemin local. Si le serveur distant interprète le fichier comme du code PHP, des variables peuvent être transmises au serveur distant via l'URL et la méthode GET. Ce n'est pas, à strictement parler, la même chose que d'hériter du contexte de variable. Le fichier inclus est en fait un script exécuté à distance, et son résultat est inclus dans le code courant.

Attention

Les versions Windows de PHP antérieures à la version 4.3.0 ne supportent pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé.

Utiliser include via HTTP

```
<?php

/* Cet exemple suppose que www.example.com est configuré pour traiter
 * les fichiers .php et non pas les fichiers .txt. De plus,
 * 'Work' signifie ici que les variables
 * $foo et $bar sont disponibles dans le fichier inclus
 */

// Ne fonctionne pas : file.txt n'a pas été traité par www.example.com comme du PHP
include 'http://www.example.com/file.txt?foo=1&bar=2';

// Ne fonctionne pas : le script cherche un fichier nommé
// 'file.php?foo=1&bar=2' sur le système local
include 'file.php?foo=1&bar=2';

// Réussi
include 'http://www.example.com/file.php?foo=1&bar=2';

$foo = 1;
$bar = 2;
include 'file.txt'; // OK.
include 'file.php'; // OK.
?>
```

Attention

Alerte de sécurité

Un fichier distant peut être traité sur le serveur distant (dépendamment de l'extension du fichier et si le serveur distant exécute PHP ou non) mais il doit toujours produire un script PHP valide parce qu'il sera traité sur le serveur local. Si le fichier du serveur distant doit être traité sur place et affiché seulement, readfile est une fonction beaucoup plus appropriée. Autrement, vous devriez bien faire attention à sécuriser le script distant afin qu'il produise un code valide et désiré.

Voir aussi travailler avec les fichiers distants, fopen et file pour des informations reliées.

Comme include et require sont des structures du langage particulières, vous devez les placer dans un bloc s'ils sont placés dans un bloc conditionnel.

include et les conditions

```
<?php

// Ceci est FAUX et échouera
if ($condition)
    include $file;
else
    include $other;
```

```
// Ceci est CORRECT.
if ($condition) {
    include $file;
} else {
    include $other;
}

?>
```

Gestion du retour : il est possible d'exécuter une commande return dans un fichier inclus pour en terminer le traitement et retourner au fichier appelant. De plus, il est possible de retourner des valeurs des fichiers inclus. Vous pouvez prendre et traiter la valeur retournée par la fonction, comme toute autre fonction. Ce n'est cependant pas possible lors de l'inclusion de fichier distant à moins que le fichier distant a des balises valides de début et de fin de script PHP (comme avec les fichiers locaux). Vous pouvez déclarer les variables nécessaire dans ces tags et elles seront introduites à l'endroit où le fichier a été inclus.

Comme include est une structure de langage particulière, les parenthèses ne sont pas nécessaires autour de l'argument. Faites attention lorsque vous comparez la valeur retournée.

Comparaison de la valeur de retour d'un include

```
<?php
// ne fonctionne pas, évaluer comme include(('vars.php') == 'OK'), i.e. include('')
if (include('vars.php') == 'OK') {
    echo 'OK';
}

// fonctionne
if ((include 'vars.php') == 'OK') {
    echo 'OK';
}

?>
```

Note

En PHP 3, le return n'est pas utilisable à moins que ce ne soit dans un bloc de fonction, auquel cas le return s'applique à une fonction et non pas au fichier.

include et return

```
return.php
<?php

$var = 'PHP';

return $var;

?>

noreturn.php
<?php

$var = 'PHP';

?>

testreturns.php
<?php

$foo = include 'return.php';
```

```

echo $foo; // affiche 'PHP'

$bar = include 'noreturn.php';

echo $bar; // affiche 1

?>

```

\$bar a la valeur de 1 car l'inclusion était réussie. Notez la différence entre les deux exemples ci-dessus. Le premier utilise la commande return dans le fichier inclus, alors que le second ne le fait pas. Si le fichier ne peut être inclus, FALSE est retourné et une erreur de niveau E_WARNING est envoyée.

S'il y a des fonctions définies dans le fichier inclus, elles peuvent être utilisées dans le fichier principal si elles sont avant le return ou après. Si le fichier est inclus deux fois, PHP 5 enverra une erreur fatale car les fonctions seront déjà déclarées, tandis que PHP 4 ne se plaindra pas des fonctions définies après return . Il est recommandé d'utiliser include_once au lieu de vérifier si le fichier a déjà été inclus et donc de retourner conditionnellement l'inclusion du fichier.

Une autre façon d'inclure un fichier PHP dans une variable est de capturer la sortie en utilisant les fonctions de contrôle de sortie avec include . Par exemple :

Utilisation de la sortie du buffer pour inclure un fichier PHP dans une chaîne

```

<?php
$string = get_include_contents('somefile.php');

function get_include_contents($filename) {
    if (is_file($filename)) {
        ob_start();
        include $filename;
        $contents = ob_get_contents();
        ob_end_clean();
        return $contents;
    }
    return false;
}

?>

```

Pour automatiquement inclure des fichiers dans vos scripts, voyez également les options de configuration auto_prepend_file et auto_append_file du php.ini .

Note

Comme ceci est une structure du langage, et non pas une fonction, il n'est pas possible de l'appeler avec les variables fonctions .

Voir aussi require , require_once , include_once , readfile , virtual et include_path .

5.7.16 require_once

La commande require_once se remplace elle-même par le fichier spécifié, un peu comme les commandes de préprocesseur C #include , et ressemble sur ce point à require . La principale différence est qu'avec require_once , vous êtes assurés que ce code ne sera ajouté qu'une seule fois, évitant de ce fait les redéfinitions de variables ou de fonctions, génératrices d'alertes.

require_once doit être utilisée dans les cas où le même fichier risque d'être inclus et évalué plusieurs fois durant l'exécution d'un script. Vous serez alors sûrs qu'il ne sera inclus qu'une fois et vous éviterez les problèmes de re-déclarations de fonctions ou autres.

Pour des exemples avec `require_once` et `include_once`, reportez-vous au code `PEAR` inclus dans les dernières distributions PHP.

Les valeurs retournées sont les mêmes que lors de l'utilisation de `include`. Si le fichier a déjà été inclus, cette fonction retourne TRUE.

Note
<code>require_once</code> a été ajouté en PHP 4.0.1pl2

Note
Attention, <code>require_once</code> et <code>include_once</code> sont insensibles à la casse sous les systèmes comme Windows.

<code>require_once</code> est insensible à la casse sur Windows
<pre>require_once("a.php"); // ceci inclut le fichier a.php require_once("A.php"); // ceci inclut encore le fichier a.php sous Windows! (uniquement en PHP 4)</pre>
Ce comportement a changé en PHP 5 - le chemin est normalisé d'abord, donc, le fichier C:\PROGRA~1\A.php est reconnu comme étant identique au fichier C:\Program Files\a.php et le fichier ne sera requis qu'une seule fois.

Attention
Les versions Windows de PHP antérieures à la version 4.3.0 ne supportent pas l'accès aux fichiers distants avec cette fonction, même si <code>allow_url_fopen</code> est activé.
Voir aussi <code>require</code> , <code>include</code> , <code>include_once</code> , <code>get_required_files</code> , <code>get_included_files</code> , <code>readfile</code> et <code>virtual</code> .

5.7.17 include_once

La commande `include_once` inclut et évalue le fichier spécifié durant l'exécution du script. Le comportement est similaire à `include`, mais la différence est que si le code a déjà été inclus, il ne le sera pas une seconde fois.

La fonction `include_once` est utilisée de préférence lorsque le fichier doit être inclus ou évalué plusieurs fois dans un script, ou bien lorsque vous voulez être sûr qu'il ne sera inclus qu'une seule fois, pour éviter des redéfinitions de fonction.

Pour plus d'exemples avec `require_once` et `include_once`, jetez un oeil dans le code de `PEAR` inclus dans la dernière distribution de PHP.

Les valeurs retournées sont les mêmes que lors de l'utilisation de `include`. Si le fichier a déjà été inclus, cette fonction retourne TRUE.

Note
<code>include_once</code> a été ajouté en PHP 4.0.1pl2

Note
Attention, <code>require_once</code> et <code>include_once</code> sont insensibles à la casse sous les systèmes comme Windows.

<code>include_once</code> est insensible à la casse sous Windows
<pre><?php include_once("a.php"); // ceci inclut le fichier a.php include_once("A.php"); // ceci inclut encore le fichier a.php sous Windows! (uniquement en PHP 4)</pre>

```
?>
```

Ce comportement a changé en PHP 5 - le chemin est normalisé d'abord, donc, le fichier C:\PROGRA~1\A.php est reconnu comme étant identique au fichier C:\Program Files\a.php et le fichier ne sera inclus qu'une seule fois.

Attention

Les versions Windows de PHP antérieures à la version 4.3.0 ne supportent pas l'accès aux fichiers distants avec cette fonction, même si `allow_url_fopen` est activé.

Voir aussi `require` , `include` , `require_once` , `get_required_files` , `get_included_files` , `readfile` et `virtual` .

5.8 Les fonctions

5.8.1 Les fonctions

Une fonction peut être définie en utilisant la syntaxe suivante :

Définition d'une fonction

```
<?php
function foo($arg_1, $arg_2, /* ..., */ $arg_n)
{
    echo "Exemple de fonction.\n";
    return $retval;
}
?>
```

Tout code PHP, correct syntaxiquement, peut apparaître dans une fonction et dans une définition de classe .

Les noms de fonctions suivent les mêmes règles que les autres labels en PHP. Un nom de fonction valide commence par une lettre ou un underscore, suivi par un nombre quelconque de lettres, de nombres ou d'underscores. Ces règles peuvent être représentées par l'expression rationnelle suivante : `[a-zA-Z_\x7f-\xff][a-zA-Z0-9_\x7f-\xff]*` .

En PHP 3, les fonctions doivent être définies avant qu'elles ne soient utilisées. Ce n'est plus le cas depuis PHP 4. **SAUF** lorsqu'une fonction est définie conditionnellement, comme montré dans les deux exemples suivants.

Lorsqu'une fonction est définie de manière conditionnelle, comme dans les exemples ci-dessous, leur définition doit **précéder** leur utilisation.

Fonctions conditionnelles

```
<?php

$makefoo = true;

/* Impossible d'appeler foo() ici,
   car cette fonction n'existe pas.
   Mais nous pouvons utiliser bar() */

bar();

if ($makefoo) {
    function foo()
```

```

    {
        echo "Je n'existe pas tant que le programme n'est pas passé ici.\n";
    }
}

/* Maintenant, nous pouvons appeler foo()
   car $makefoo est maintenant vrai */

if ($makefoo) foo();

function bar()
{
    echo "J'existe dès de début du programme.\n";
}

?>

```

Fonctions dans une autre fonction

```

<?php
function foo()
{
    function bar()
    {
        echo "Je n'existe pas tant que foo() n'est pas appelé.\n";
    }
}

/* Impossible d'appeler bar() ici
   car il n'existe pas. */

foo();

/* Maintenant, nous pouvons appeler bar(),
   car l'utilisation de foo() l'a rendu
   accessible. */

bar();

?>

```

Toutes les fonctions et classes en PHP ont une portée globale - elles peuvent être appelées à l'extérieur d'une fonction si elles ont été définies à l'intérieur et vis-versa.

PHP ne supporte pas le surchargement de fonction, ni la destruction ou la redéfinition de fonctions déjà déclarées.

Note

Les noms de fonctions sont insensibles à la casse, et il est généralement admis que les fonctions doivent être appelée avec le nom utilisé dans leur déclaration, y compris la casse.

PHP 3 ne supporte pas un nombre variable d'arguments (voir [valeurs par défaut d'arguments](#) pour plus d'informations). Les deux sont supportés, depuis PHP 4 : voir [liste variable d'arguments de fonction](#) et les fonctions de références que sont [func_num_args](#) , [func_get_arg](#) , et [func_get_args](#) pour plus d'informations.

Il est possible d'appeler des fonctions récursives en PHP. Cependant, les appels de méthodes/fonctions récursives avec 100-200 degrés de récursion peuvent remplir la pile et ainsi, terminer le script courant.

Fonctions récursives

```
<?php
function recursion($a)
{
    if ($a < 20) {
        echo "$a\n";
        recursion($a + 1);
    }
}
?>
```

5.8.2 Les arguments de fonction

Des informations peuvent être passées à une fonction en utilisant une liste d'arguments, dont chaque expression est séparée par une virgule.

PHP supporte le passage d'arguments par valeur (méthode par défaut), par référence. Les listes variables d'arguments sont supportées par PHP 4 et les versions plus récentes. Voir liste variable d'arguments de fonction et les fonctions utiles que sont func_num_args, func_get_arg, et func_get_args. On peut arriver au même résultat en PHP 3 en passant un tableau comme argument de la fonction :

Nombre variable d'argument sous forme de tableau

```
<?php
function takes_array($input)
{
    echo "$input[0] + $input[1] = ", $input[0]+$input[1];
}
?>
```

5.8.2.1 Passage d'arguments par référence

Par défaut, les arguments sont passés à la fonction par valeur (donc vous pouvez changer la valeur d'un argument dans la fonction, cela ne change pas sa valeur à l'extérieur de la fonction). Si vous voulez que vos fonctions puissent changer la valeur des arguments, vous devez passer ces arguments par référence.

Si vous voulez qu'un argument soit toujours passé par référence, vous pouvez ajouter un '&' devant l'argument dans la déclaration de la fonction :

Passage d'arguments par référence

```
<?php
function add_some_extra(&$string)
{
    $string .= ', et un peu plus.';
}
$str = 'Ceci est une chaîne';
add_some_extra($str);
echo $str;    // affiche 'Ceci est une chaîne, et un peu plus.'
?>
```

5.8.2.2 Valeur par défaut des arguments

Vous pouvez définir comme en C++ des valeurs par défaut pour les arguments de type scalaire :

Valeur par défaut des arguments de fonctions

```
<?php
function servir_cafe ($type = "cappuccino")
{
    return "Servir un $type.\n";
}
echo servir_cafe();
echo servir_cafe("espresso");
?>
```

L'exemple ci-dessus va afficher :

Servir un cappuccino.
Servir un espresso.

PHP vous autorise à utiliser des tableaux ainsi que le type spécial NULL comme valeur par défaut, par exemple :

Utilisation de type non scalaire comme valeur par défaut

```
<?php
function servir_cafe($types = array("cappuccino"), $coffeeMaker = NULL)
{
    $device = is_null($coffeeMaker) ? "les mains" : $coffeeMaker;
    return "Préparation d'une tasse de ".join(", ", $types)." avec $device.\n";
}
echo servir_cafe();
echo servir_cafe(array("cappuccino", "lavazza"), "une cafetière");
?>
```

La valeur par défaut d'un argument doit obligatoirement être une constante, et ne peut être ni une variable, ni un membre de classe, ni un appel de fonction.

Il est à noter que si vous utilisez des arguments avec valeur par défaut avec d'autres sans valeur par défaut, les premiers doivent être placés à la suite de tous les paramètres sans valeur par défaut. Sinon, cela ne fonctionnera pas. Considérons le code suivant :

Les arguments à valeur par défaut doivent être en premier : erreur

```
<?php
function faireunyaourt ($type = "acidophilus", $flavour)
{
    return "Préparer un bol de $type $flavour.\n";
}

echo faireunyaourt("framboise"); // ne fonctionne pas comme voulu
?>
```

L'exemple ci-dessus va afficher :

Warning: Missing argument 2 in call to makeyogurt() in
/usr/local/etc/httpd/htdocs/php3test/functest.html on line 41
Préparer un bol de framboise.

Maintenant comparons l'exemple précédent avec l'exemple suivant :

Les arguments à valeur par défaut doivent être en premier : valide

```
<?php
function faireunyaourt ($flavour, $type = "acidophilus")
{
```

```

    return "Préparer un bol de $type $flavour.\n";
}
echo faireunyaourt ("framboise"); // fonctionne comme voulu
?>

```

L'exemple ci-dessus va afficher :

Préparer un bol de acidophilus framboise.

Note

Depuis PHP 5, les valeurs par défaut peuvent être passées par référence.

5.8.2.3 Nombre d'arguments variable

PHP 4 et suivants supportent les fonctions à nombre d'arguments variable. C'est très simple à utiliser, avec les fonctions `func_num_args`, `func_get_arg` et `func_get_args`.

Aucune syntaxe particulière n'est nécessaire, et la liste d'argument doit toujours être fournie explicitement avec la définition de la fonction, et se comportera normalement.

5.8.3 Les valeurs de retour

Les valeurs sont renvoyées en utilisant une instruction de retour optionnelle. Tous les types de variables peuvent être renvoyés, tableaux et objets compris. Cela fait que la fonction finit son exécution immédiatement et passe le contrôle à la ligne appelante. Voir [return](#) pour plus d'informations.

Utilisation de `return`

```

<?php
function carre ($num)
{
    return $num * $num;
}
echo carre (4); // affiche '16'.
?>

```

Vous ne pouvez pas renvoyer plusieurs valeurs en même temps, mais vous pouvez obtenir le même résultat en renvoyant un tableau.

Retourner un tableau d'une fonction

```

<?php
function petit_nombre()
{
    return array (0, 1, 2);
}
list ($zero, $one, $two) = petit_nombre();
?>

```

Pour retourner une référence d'une fonction, utilisez l'opérateur `&` aussi bien dans la déclaration de la fonction que dans l'assignation de la valeur de retour.

Retourner une référence d'une fonction

```

<?php
function &retourne_reference()

```

```

{
    return $uneref;
}

$newref =& retourne_reference();
?>

```

Pour plus d'informations sur les références, référez-vous à [l'explication sur les références](#) .

5.8.4 Fonctions variables

PHP supporte le concept de fonctions variables. Cela signifie que si le nom d'une variable est suivi de parenthèses, PHP recherchera une fonction de même nom, et essaiera de l'exécuter. Cela peut servir, entre autre, pour faire des fonctions callback, des tables de fonctions...

Les fonctions variables ne peuvent pas fonctionner avec les éléments de langage comme les [echo](#) , [print](#) , [unset](#) , [isset](#) , [empty](#) , [include](#) , [require](#) etc. Vous devez utiliser votre propre gestion de fonctions pour utiliser un de ces éléments de langages comme fonctions variables.

Exemple de fonction variable

```

<?php
function foo() {
    echo "dans foo()<br />\n";
}

function bar($arg = '' )
{
    echo "Dans bar(); l'argument était '$arg'.<br />\n";
}

// Ceci est une fonction détourné de echo
function echoit($string)
{
    echo $string;
}

$func = 'foo';
$func();          // Appel foo()

$func = 'bar';
$func( 'test' ); // Appel bar()

$func = 'echoit';
$func('test');  // Appel echoit()
?>

```

Vous pouvez aussi appeler une méthode d'un objet en utilisant le système des fonctions variables.

Exemple de méthode variable

```

<?php
class Foo
{
    function Variable()
    {
        $name = 'Bar';
        $this->$name(); // Appel la méthode Bar()
    }

    function Bar()
    {

```

```

        echo "C'est Bar";
    }
}

$foo = new Foo();
$funcname = "Variable";
$foo->$funcname(); // Appel $foo->Variable()

?>

```

Voir aussi [call_user_func](#) , [les variables variables](#) et [function_exists](#) .

5.8.5 Fonctions internes

PHP dispose de nombreuses fonctions et structures standards. Il y a aussi des fonctions qui requièrent des extensions spécifiques de PHP, sans lesquelles vous obtiendrez l'erreur fatale `undefined function` . Par exemple, pour utiliser les fonctions [d'images](#) , telles que [imagecreatetruecolor](#) , vous aurez besoin du support de GD dans PHP. Ou bien, pour utiliser [mysql_connect](#) , vous aurez besoin de l'extension [MySQL](#) . Il y a des fonctions de base qui sont incluses dans toutes les versions de PHP, telles que les fonctions de [chaînes de caractères](#) et les fonctions de [variables](#) . Utilisez [phpinfo](#) ou [get_loaded_extensions](#) pour savoir quelles sont les extensions qui sont compilées avec votre PHP. Notez aussi que de nombreuses extensions sont activées par défaut, et que le manuel PHP est compartimenté par extension. Voyez les chapitres de [configuration](#) , [installation](#) ainsi que les détails particuliers à chaque extension, pour savoir les mettre en place.

Lire et comprendre le prototype d'une fonction est décrit dans l'annexe [Comment lire la définition d'une fonction \(prototype\)](#) . Il est important de comprendre ce qu'une fonction retourne, ou si une fonction travaille directement sur la valeur des paramètres fournis. Par exemple, [str_replace](#) va retourner une chaîne modifiée, tandis que [usort](#) travaille directement sur la variable passée en paramètre. Chaque page du manuel a des informations spécifiques sur chaque fonction, comme le nombre de paramètres, les évolutions de spécifications, les valeurs retournées en cas de succès ou d'échec, et la disponibilité en fonction des versions. Bien connaître ces différences, parfois subtiles, est crucial pour bien programmer en PHP.

Voir aussi [function_exists](#) , [l'index des fonctions](#) , [get_extension_funcs](#) et [dl](#) .

5.9 Les classes et les objets (PHP 4)

5.9.1 Les classes : class

Une classe est une collection de variables et de fonctions qui fonctionnent avec ces variables. Une classe est définie en utilisant la syntaxe suivante :

Les classes en PHP

```

<?php
class Panier {
    // Eléments de notre panier
    var $items;

    // Ajout de $num articles de type $artnr au panier

    function add_item ($artnr, $num) {

```

```

        $this->items[$artnr] += $num;
    }

    // Suppression de $num articles du type $artnr du panier

    function remove_item($artnr, $num) {
        if ($this->items[$artnr] > $num) {
            $this->items[$artnr] -= $num;
            return true;
        } elseif ($this->items[$artnr] == $num) {
            unset($this->items[$artnr]);
            return true;
        } else {
            return false;
        }
    }
}
?>

```

L'exemple ci-dessus définit la classe Panier qui est composée d'un tableau associatif contenant les articles du panier et de deux fonctions, une pour ajouter et une pour enlever des éléments au panier.

Attention

Vous **NE POUVEZ PAS** couper la définition d'une classe en plusieurs fichiers. De la même façon, vous **NE POUVEZ PAS** couper la définition d'une classe en de multiples blocs, à moins que la coupure ne soit à l'intérieure de la déclaration d'une méthode. Ce qui suit ne fonctionnera pas :

Ne pas couper la définition d'une classe

```

<?php
class test {
?>
<?php
    function test() {
        echo 'OK';
    }
}
?>

```

Néanmoins, ce qui suit est autorisé :

Changer de mode durant la définition d'une classe

```

<?php
class test {
    function test() {
        ?>
        <?php
            print 'Ok';
        }
    }
}
?>

```

Les notes suivantes ne sont valables que pour PHP 4.

Attention

Le nom stdClass est utilisé en interne par Zend et ne doit pas être utilisé. Vous ne pouvez pas nommer une classe stdClass en PHP.

Attention

Les noms de fonctions `__sleep` et `__wakeup` sont magiques en PHP. Vous ne pouvez pas utiliser ces noms de fonctions dans vos classes, à moins que vous ne souhaitiez utiliser la magie qui y est associée.

Attention

PHP se réserve l'usage de tous les noms de fonctions commençant par `__`, pour sa propre magie. Il est vivement recommandé de ne pas utiliser des noms de fonctions commençant par `__`, à moins que vous ne souhaitiez utiliser la magie qui y est associée.

Note

En PHP 4, seuls les initialiseurs constants pour les variables `var` sont autorisés. Utilisez les constructeurs pour les initialisations variables, ou utilisant des expressions.

Initialisation de membres de classe

```
<?php
/* Aucune de ces syntaxes ne fonctionnera en PHP 4 */
class Panier {
    var $date_du_jour = date("d/m/Y");
    var $name = $firstname;
    var $owner = 'Fred ' . 'Jones';
    var $items = array("DVD", "Télé", "Magnétoscope");
}
/* Voici comment cela doit se faire désormais. */
class Panier {
    var $date_du_jour;
    var $name;
    var $owner;
    var $items;
    function Panier() {
        $this->date_du_jour = date("d/m/Y");
        $this->name = $GLOBALS['firstname'];
        /* etc. */
    }
}
?>
```

Les classes forment un type de variable. Pour créer une variable du type désiré, vous devez utiliser l'opérateur `new`.

Création d'un objet de classe

```
<?php
$cart = new Panier;
$cart->add_item("10", 1);
$another_cart = new Panier;
$another_cart->add_item("0815", 3);
?>
```

L'instruction ci-dessus crée l'objet `$cart` de la classe `Panier`. La fonction `add_item()` est appelée afin d'ajouter l'article numéro 10 dans le panier. 3 articles numéro 0815 sont ajoutés au panier `$another_cart`.

`$cart` et `$another_cart` disposent des fonctions `add_item()`, `remove_item()` et de la variable `items`. Ce sont des fonctions et variables distinctes. Vous pouvez vous représenter les objets comme des dossiers sur votre disque dur. Vous pouvez avoir deux fichiers `lisez-moi.txt` sur votre disque dur, tant qu'ils ne sont pas dans le même répertoire. De même que vous devez alors taper le chemin complet jusqu'au fichier, vous devez spécifier le nom complet de la méthode avant de l'employer : en termes PHP, le dossier racine est l'espace de nom global, et le séparateur de dossier est `->`. Par exemple, les noms `$cart->items` et `$another_cart->items` représentent deux variables distinctes. Notez que le nom de la variable est alors `$cart->items`, et non pas `$cart->$items` : il n'y a qu'un seul signe `$` dans un nom de variable.

Accès aux membres d'une classe

```
<?php
// correct, le signe $ est unique
$cart->items = array("10" => 1);
// incorrect, car $cart->$items devient $cart->""
$cart->$items = array("10" => 1);
// correct, mais risque de ne pas se comporter comme prévu
// $cart->$myvar devient $cart->items
$myvar = 'items';
$cart->$myvar = array("10" => 1);
?>
```

A l'intérieur d'une définition de classe, vous ne savez pas le nom de la variable à partir duquel l'objet sera accessible dans le script. On ne peut prévoir que l'objet créé sera affecté à la variable `$cart` ou `$another_cart`. Donc, vous ne pouvez pas utiliser la syntaxe `$cart->items`. Mais pour pouvoir accéder à aux méthodes et membres d'un objet, vous pouvez utiliser la variable spéciale `$this`, qui peut s'interpréter comme "moi-même", ou bien "l'objet courant". Par exemple, `'$this->items[$artnr] += $num;'` peut se lire comme 'ajouter `$num` au compteur `$artnr` de mon propre tableau de compteur' ou bien 'ajouter `$num` au compteur `$artnr` du tableau de compteurs de l'objet courant'.

Note

La pseudo-variable `$this` n'est pas toujours définie si la méthode dans laquelle elle est présente est appelée statiquement. Cependant, ceci n'est pas une règle stricte : `$this` est définie si une méthode est appelée statiquement depuis un autre objet. Dans ce cas, sa valeur vaut l'objet appelé. Ce comportement est illustré dans l'exemple ci-dessous :

`$this` peut être indéfinie : cas d'une méthode statique

```
<?php
class A
{
    function foo()
    {
        if (isset($this)) {
            echo '$this est défini (';
            echo get_class($this);
            echo ")\n";
        } else {
            echo "\$this n'est pas défini.\n";
        }
    }
}

class B
{
    function bar()
    {
        A::foo();
    }
}

$a = new A();
$a->foo();
A::foo();
$b = new B();
$b->bar();
B::bar();
?>
```

L'exemple ci-dessus va afficher :

```

$this est défini (a)
$this n'est pas défini.
$this est défini (b)
$this n'est pas défini.

```

Note

Il y a des fonctions très pratiques pour gérer les classes et objets. Vous pouvez étudier le chapitre sur les [fonctions de classes et objets](#) .

5.9.2 extends : héritage

Souvent, vous aurez besoin d'une classe avec des méthodes et fonctions similaires à une autre classe. En fait, il est bon de définir des classes génériques, qui pourront être réutilisées et adaptées à tous vos projets. Pour faciliter cela, une classe peut être une extension d'une autre classe. La classe dérivée hérite alors de toutes les méthodes et variables de la classe de base (cet héritage a de bien que personne ne meurt pour en profiter), mais peut définir ses propres fonctions et variables, qui s'ajouteront. Une classe ne peut hériter que d'une seule autre classe, et l'héritage multiple n'est pas supporté. Les héritages se font avec le mot clé ' extends ' .

Héritage de classes

```

<?php
class Panier_nomme extends Panier {
    var $owner;
    function set_owner ($name) {
        $this->owner = $name;
    }
}
?>

```

L'exemple ci-dessus définit la classe Panier_nomme qui possède les mêmes variables que la classe Panier et la variable \$owner en plus, ainsi que la fonction set_owner() . Vous créez un panier nominatif de la même manière que précédemment, et vous pouvez alors affecter un nom au panier ou en connaître le nom. Vous pouvez de toutes les façons utiliser les mêmes fonctions que sur un panier classique.

Héritage de classes (2)

```

<?php
$ncart = new Panier_nomme; // Création d'un panier nominatif
$ncart->set_owner ("kris"); // Affectation du nom du panier
echo $ncart->owner; // Affichage du nom du panier
$ncart->add_item ("10", 1); // (héritage des fonctions de la classe père)
?>

```

5.9.3 Constructeur

Le constructeur est la fonction qui est appelée automatiquement par la classe lorsque vous créez une nouvelle instance d'une classe à l'aide de l'opérateur new . La fonction constructeur a le même nom que la classe. Une fonction devient le constructeur si elle porte le même nom que la classe. Si une classe n'a pas de constructeur, le constructeur de la classe de base est appelé, si il existe.

Constructeur de classe

```

<?php
class Auto_Panier extends Panier {
    function Auto_Panier () {
        $this->add_item ("10", 1);
    }
}

```

```

    }
}
?>

```

L'exemple ci-dessus définit la classe `Auto_Panier` qui hérite de la classe `Panier` et définit le constructeur de la classe. Ce dernier initialise le panier avec 1 article de type numéro 10 dès que l'instruction `new` est appelée. La fonction constructeur peut prendre ou non des paramètres optionnels, ce qui la rend beaucoup plus pratique. Pour pouvoir utiliser cette classe sans paramètre, tous les paramètres du constructeurs devraient être optionnels, en fournissant une valeur par défaut, comme ci-dessous.

Constructeur de classe à argument optionnel

```

<?php
class Constructor_Cart extends Cart {
    function Constructor_Cart ($item = "10", $num = 1) {
        $this->add_item ($item, $num);
    }
}
// Création du Panier
$default_cart = new Constructor_Cart;
// Création d'un vrai Panier
$different_cart = new Constructor_Cart ("20", 17);
?>

```

Vous pouvez également utiliser l'opérateur `@` pour **empêcher** les erreurs survenant dans le constructeur de s'afficher, e.g. `@new`.

Constructeur de classe

```

<?php
class A {
    function A() {
        echo "Je suis le constructeur de A.<br />\n";
    }
    function B() {
        echo "Je suis une fonction standard appelée B dans la classe A.<br />\n";
        echo "Je ne suis pas le constructeur de A.<br />\n";
    }
}

class B extends A {
}
// Cette syntaxe va appeler B() comme constructeur.
$b = new B;
?>

```

La fonction `B()` de la classe `A` va soudainement devenir le constructeur de la classe `B`, bien qu'il n'ait pas été prévu pour. PHP 4 ne se soucie guère si la fonction est définie dans la classe `B` ou si elle a été héritée.

Attention

PHP 4 n'appelle pas automatiquement le constructeur de la classe supérieure depuis le constructeur de la classe dérivée. Il est de votre responsabilité de propager l'appel des constructeurs.

Les destructeurs sont des fonctions qui sont appelées lorsqu'un objet est détruit, soit avec la fonction `unset` soit par simple sortie d'une fonction (cas des variables locales). Il n'y a pas de destructeurs en PHP. Vous devez utiliser la fonction `register_shutdown_function` à la place pour simuler la plus part des effets des destructeurs.

5.9.4 Opérateur de contexte de classe (::)

Attention

La documentation suivante n'est valable que pour PHP 4 et plus récent.

Parfois, il est pratique de faire référence aux fonctions et variables d'une classe de base, ou bien d'utiliser des méthodes de classes qui n'ont pas encore d'objets créés. L'opérateur :: est là pour ces situations.

Exemples avec l'opérateur ::

```
<?php
class A {
    function example() {
        echo "Je suis la fonction originale A::example().<br />\n";
    }
}
class B extends A {
    function example() {
        echo "Je suis la fonction redéfinie B::example().<br />\n";
        A::example();
    }
}
// Il n'y a pas d'objets de classe A.
// L'affichage est :
// Je suis la fonction originale A::example().<br />
A::example();
// Création d'un objet de la classe B.
$b = new B;
// L'affichage est :
// Je suis la fonction redéfinie B::example().<br />
// Je suis la fonction originale A::example().<br />
$b->example();
?>
```

Les exemples ci-dessus appellent la fonction `example()` dans la classe A, mais il n'y a pas encore d'objet de classe A, alors il n'est pas possible d'écrire `$a->example()`. A la place, on appelle la fonction `example()` comme une fonction de classe, c'est-à-dire avec le nom de la classe elle-même, et sans objet.

Il y a des fonctions de classe, mais pas de variables de classe. En fait, il n'y a aucun objet au moment de l'appel de la fonction. Donc, une fonction de classe ne peut accéder à aucune variable (mais elle peut accéder aux variables locales et globales). Il faut proscrire l'utilisation de `$this`.

Dans l'exemple ci-dessus, la classe B redéfinit la fonction `example()`. La définition originale dans la classe A est remplacée par celle de B, et n'est plus accessible depuis B, à moins que vous n'appeliez spécifiquement la fonction `example()` de la classe A avec l'opérateur ::. Ecrivez `A::example()` pour cela (en fait, il faudrait écrire `parent::example()`, comme expliqué dans la section suivante).

Dans ce contexte, il y a un objet courant, qui peut avoir d'autres variables objets. De ce fait, lorsqu'il est utilisé depuis une méthode d'un objet, vous pouvez utiliser `$this`.

5.9.5 parent

Il arrive que vous ayez à écrire du code qui fait référence aux variables et fonctions des classes de base. C'est particulièrement vrai si votre classe dérivée est une spécialisation de votre classe de base.

Au lieu d'utiliser le nom littéral de votre classe de base dans votre code, vous pouvez utiliser le mot réservé `parent`, qui représente votre classe de base (celle indiquée par `extends`, dans la déclaration de votre classe). En faisant cela, vous évitez d'appeler le nom de votre classe de base directement dans votre code. Si votre héritage change, vous n'aurez plus qu'à modifier le nom de la classe dans la déclaration `extends` de votre classe.

Exemples avec l'opérateur parent

```
<?php
class A {
    function example() {
        echo "Je suis A::example() et je fournis une fonctionnalité de base.<br />\n";
    }
}
class B extends A {
    function example() {
        echo "Je suis B::example() et je fournis une fonctionnalité supplémentaire.<br />\n";
        parent::example();
    }
}
$b = new B;
// Cette syntaxe va appeler B::example(), qui, à son tour, va appeler A::example().
$b->example();
?>
```

5.9.6 Sauvegarde d'objets - cas des sessions

Note

En PHP 3, les objets perdent leur association de classe à travers le processus de sauvegarde et relecture. Le type de la variable après relecture est bien objet mais il n'a plus de méthode ou de nom de classe. Cela rend la fonctionnalité plutôt inutile (l'objet est devenu un tableau avec une syntaxe étrange).

Attention

La documentation suivante n'est valable que pour PHP ≥ 4 .

`serialize` retourne une chaîne représentant une valeur qui peut être stockée dans les sessions de PHP, ou une base de données. `unserialize` peut relire cette chaîne pour recréer la valeur originale. `serialize` va sauvegarder toutes les variables d'un objet. Le nom de la classe sera sauvegardé mais pas les méthodes de cet objet.

Pour permettre à `unserialize` de lire un objet, la classe de cet objet doit être définie. C'est-à-dire, si vous avez un objet `$a` de la classe `A` dans une page `php1.php`, et que vous le linéarisez avec `serialize`, vous obtiendrez une chaîne qui fait référence à la classe `A`, et contient toutes les valeurs de `$a`. Pour pouvoir le relire avec la fonction `unserialize` dans une page `page2.php`, il faut que la définition de la classe `A` soit présente dans cette deuxième page. Cela peut se faire de manière pratique en sauvegardant la définition de la classe `A` dans un fichier séparé, et en l'incluant dans les deux pages `page1.php` et `page2.php`.

La classe d'un objet doit être définie avant sa délinéarisation

```
<?php
classa.inc:
class A {
    var $one = 1;
    function show_one() {
        echo $this->one;
    }
}
?>
page1.php:
```

```

<?php
    include("classa.inc");
    $a = new A;
    $s = serialize($a);
    // enregistrez $s où la page2.php pourra le trouver.
    $fp = fopen("store", "w");
    fputs($fp, $s);
    fclose($fp);
?>
page2.php:
<?php
    // Ceci est nécessaire pour que unserialize() fonctionne correctement
    include("classa.inc");
    $s = implode("", @file("store"));
    unserialize($s);
    // maintenant, utilisez la méthode show_one de l'objet $a.
    $a->show_one();
?>

```

Si vous utilisez les sessions et la fonction `session_register` pour sauver des objets, ces objets seront linéarisés automatiquement avec la fonction `serialize` à la fin de chaque script, et relus avec `unserialize` au début du prochain script. Cela signifie que ces objets peuvent apparaître dans n'importe quelle page qui utilise vos sessions.

Il est vivement recommandé d'inclure la définition de classe dans toutes vos pages, même si vous n'utilisez pas ces classes dans toutes vos pages. Si vous l'oubliez et qu'un tel objet est présent, il perdra sa classe, et deviendra un objet de classe `stdClass` sans aucune fonction, et donc, plutôt inutile.

Si, dans l'exemple ci-dessus, `$a` devient un objet de session avec l'utilisation de `session_register("a")`, vous devez penser à inclure le fichier `classa.inc` dans toutes vos pages, et pas seulement `page1.php` et `page2.php`.

5.9.7 Les fonctions magiques `__sleep` et `__wakeup`

`serialize` s'assure que votre classe a une méthode avec le nom magique `__sleep`. Si c'est le cas, cette fonction est appelée avant toute linéarisation. Elle peut alors nettoyer l'objet et on s'attend à ce qu'elle retourne un tableau avec la liste des noms de variables qui doivent être sauvées.

Le but de cette fonction `__sleep` est de fermer proprement toute connexion à une base de données, de valider les requêtes, de finaliser toutes les actions commencées. Cette fonction est aussi pratique si vous avez de très grands objets qui n'ont pas besoin d'être sauvés entièrement.

A l'inverse, `unserialize` s'assure de la présence de la fonction magique `__wakeup`. Si elle existe, cette fonction reconstruit toutes les ressources d'un objet.

Le but de cette fonction `__wakeup` est de rétablir toutes les connexions aux bases de données, et de recréer les variables qui n'ont pas été sauvées.

5.9.8 Références dans un constructeur

Créer des références dans un constructeur peut conduire à des résultats étranges. Ce tutoriel vous guide pour éviter ces problèmes.

Références dans un constructeur

```

<?php
class foo {
    function foo($name) {
        // crée une référence dans le tableau global $globalref
        global $globalref;
        $globalref[] = &$this;
        // donne le nom de la variable
        $this->setName($name);
        // et l'affiche
        $this->echoName();
    }
    function echoName() {
        echo "<br />", $this->Name;
    }
    function setName($name) {
        $this->Name = $name;
    }
}
?>

```

Vérifions maintenant qu'il y a une différence entre \$bar1 qui a été créé avec = et \$bar2 qui a été créé avec l'opérateur de référence =& :

Références dans un constructeur (2)

```

<?php
$bar1 = new foo('créé dans le constructeur');
$bar1->echoName();
$globalref[0]->echoName();

/* affiche :
créé dans le constructeur
créé dans le constructeur
créé dans le constructeur */

$bar2 =&new foo('créé dans le constructeur');
$bar2->echoName();
$globalref[1]->echoName();
/* affiche :
créé dans le constructeur
créé dans le constructeur
créé dans le constructeur */
?>

```

Apparemment, il n'y a pas de différence, mais en fait, il y en a une significative : \$bar1 et \$globalref[0] ne sont pas référencées, ces deux variables sont différentes. Cela est dû au fait que l'opérateur new ne retourne pas de référence, mais retourne une copie.

Note

Il n'y a aucune perte de performances (puisque PHP 4 utilise un compteur de références) à retourner des copies au lieu de références. Au contraire, il est souvent mieux de travailler sur les copies plutôt que sur les références, car créer une référence prend un peu plus de temps que de créer une copie qui ne prend virtuellement pas de temps (à moins de créer un tableau géant ou un objet monstrueux, auquel cas il est préférable de passer par des références).

Pour prouver ceci, regardez le code suivant :

Références et constructeurs

```

<?php
// maintenant, nous allons changer de nom. Qu'attendez-vous?
// Vous pouvez vous attendre à ce que les deux variables $bar
// et $globalref[0] changent de nom...
$bar1->setName('modifié');

```

```

// comme prédit, ce n'est pas le cas
$bar1->echoName();
$globalref[0]->echoName();

/* affiche :
modifié
créé dans le constructeur
*/

// quelle est la différence entre $bar2 et $globalref[1]
$bar2->setName('modifié');

// Heureusement, elles sont non seulement égales, mais
// elles représentent la même variable.
// donc $bar2->Name et $globalref[1]->Name sont les mêmes
$bar2->echoName();
$globalref[1]->echoName();

/* affiche :
  modifié
  modifié */
?>

```

Un dernier exemple pour bien comprendre.

Références dans un constructeur (3)

```

<?php
class a {
    function a($i) {
        $this->value = $i;
        // Essayez de comprendre on n'a pas besoin de
        // référence ici
        $this->b = new b($this);
    }
    function createRef() {
        $this->c = new b($this);
    }
    function echoValue() {
        echo "<br />", "class ", get_class($this), ': ', $this->value;
    }
}
class b {
    function b(&$a) {
        $this->a = &$a;
    }
    function echoValue() {
        echo "<br />", "class ", get_class($this), ': ', $this->a->value;
    }
}
// Essayez de comprendre pourquoi une copie simple va
// conduire à un résultat indésirable à
// la ligne marquée d'une étoile
$a =&new a(10);
$a->createRef();
$a->echoValue();
$a->b->echoValue();
$a->c->echoValue();
$a->value = 11;
$a->echoValue();
$a->b->echoValue(); // *
$a->c->echoValue();

?>

```

L'exemple ci-dessus va afficher :

```
class A: 10
class B: 10
class B: 10
class A: 11
class B: 11
class B: 11
```

5.9.9 Comparer des objets

En PHP 4, les objets sont comparés de manière très simple, à savoir : deux instances sont égales si elles ont les mêmes attributs et valeurs, et qu'elles sont de la même classe. Des règles similaires s'appliquent lors de la comparaison avec l'opérateur `===`.

Si vous exécutez le code suivant :

Exemple de comparaison d'objets en PHP 4

```
<?php

function bool2str($bool) {
    if ($bool === false) {
        return 'FALSE';
    } else {
        return 'TRUE';
    }
}

function compareObjects(&$o1, &$o2) {
    echo 'o1 == o2 : '.bool2str($o1 == $o2)."\n";
    echo 'o1 != o2 : '.bool2str($o1 != $o2)."\n";
    echo 'o1 === o2 : '.bool2str($o1 === $o2)."\n";
    echo 'o1 !== o2 : '.bool2str($o1 !== $o2)."\n";
}

class Flag {
    var $flag;

    function Flag($flag=true) {
        $this->flag = $flag;
    }
}

class SwitchableFlag extends Flag {

    function turnOn() {
        $this->flag = true;
    }

    function turnOff() {
        $this->flag = false;
    }
}

$o = new Flag();
$p = new Flag(false);
$q = new Flag();

$r = new SwitchableFlag();

echo "Compare des instances créées avec les mêmes paramètres\n";
```

```

compareObjects($o, $q);

echo "\nCompare des instances créées avec différents paramètres\n";
compareObjects($o, $p);

echo "\nCompare une instance d'un parent avec celle d'une sous-classe\n";
compareObjects($o, $r);

?>

```

L'exemple ci-dessus va afficher :

Compare des instances créées avec les mêmes paramètres

```

o1 == o2 : TRUE
o1 != o2 : FALSE
o1 === o2 : TRUE
o1 !== o2 : FALSE

```

Compare des instances créées avec différents paramètres

```

o1 == o2 : FALSE
o1 != o2 : TRUE
o1 === o2 : FALSE
o1 !== o2 : TRUE

```

Compare une instance d'un parent avec celle d'une sous-classe

```

o1 == o2 : FALSE
o1 != o2 : TRUE
o1 === o2 : FALSE
o1 !== o2 : TRUE

```

Ce qui est le résultat que nous attendions, au vue des règles édictées. Seules deux instances avec les mêmes valeurs d'attributs, et issues de la même classe sont considérées comme égales.

Même lorsque nous avons une composition d'objet, la même règle de comparaison s'applique. Dans l'exemple ci-dessous, nous allons créer une classer conteneur, qui stocke un tableau associatif Flag .

Comparaison d'objets composés en PHP 4

```

<?php

class FlagSet {
    var $set;

    function FlagSet($flagArr = array()) {
        $this->set = $flagArr;
    }

    function addFlag($name, $flag) {
        $this->set[$name] = $flag;
    }

    function removeFlag($name) {
        if (array_key_exists($name, $this->set)) {
            unset($this->set[$name]);
        }
    }
}

```

```

$u = new FlagSet();
$u->addFlag('flag1', $o);
$u->addFlag('flag2', $p);
$v = new FlagSet(array('flag1'=>$q, 'flag2'=>$p));
$w = new FlagSet(array('flag1'=>$q));

echo "\nObjects composés u(o,p) et v(q,p)\n";
compareObjects($u, $v);

echo "\nu(o,p) et w(q)\n";
compareObjects($u, $w);

?>

```

L'exemple ci-dessus va afficher :

```

Objects composés u(o,p) et v(q,p)
o1 == o2 : TRUE
o1 != o2 : FALSE
o1 === o2 : TRUE
o1 !== o2 : FALSE

u(o,p) et w(q)
o1 == o2 : FALSE
o1 != o2 : TRUE
o1 === o2 : FALSE
o1 !== o2 : TRUE

```

5.10 Les classes et les objets (PHP 5)

5.10.1 Introduction

En PHP 5, il y a un tout nouveau model objet. La gestion des objets en PHP a été complètement réécrite, permettant de meilleurs performances ainsi que plus de fonctionnalités.

5.10.2 Syntaxe de base

5.10.2.1 class

Chaque définition de classe commence par le mot-clé `class`, suivi par le nom de la classe, qui peut être quelconque à condition que ce ne soit pas un mot réservé en PHP. Suivent une paire de parenthèses contenant la définition des membres et des méthodes. Une pseudo-variable `$this` est disponible lorsqu'une méthode est appelée depuis un contexte objet. `$this` est une référence à l'objet appelé (habituellement, l'objet auquel la méthode appartient, mais ce peut être un autre objet si la méthode est appelée de manière statique depuis le contexte d'un autre objet). Ce comportement est illustré dans l'exemple suivant :

La variable `$this` en programmation objet

```

<?php
class A
{

```

```

function toto()
{
    if (isset($this)) {
        echo '$this est définie (';
        echo get_class($this);
        echo ")\n";
    } else {
        echo "\$this n'est pas définie.\n";
    }
}

class B
{
    function titi()
    {
        A::toto();
    }
}

$a = new A();
$a->toto();
A::toto();
$b = new B();
$b->titi();
B::titi();
?>

```

L'exemple ci-dessus va afficher :

```

$this est définie (a)
$this n'est pas définie.
$this est définie (b)
$this n'est pas définie.

```

Définition simple d'une classe

```

<?php
class SimpleClass
{
    // déclaration d'un membre
    public $var = 'une valeur par défaut';

    // déclaration de la méthode
    public function displayVar() {
        echo $this->var;
    }
}
?>

```

5.10.2.2 Le mot clé new

Pour créer une instance d'un objet, un nouvel objet doit être créé et assigné à une variable. Un objet doit toujours être assigné lors de la création d'un nouvel objet à moins qu'un l'objet ait un constructeur défini qui lance un exception en cas d'erreur.

Création d'une instance

```

<?php
$instance = new SimpleClass();
?>

```

Lors de l'assignation d'une instance déjà créée d'un objet à une variable, la nouvelle variable accèdera à la même instance de l'objet assigné. Ce comportement est le même que lors du passage d'une instance à une fonction. Une nouvelle instance d'un objet déjà créé peut être effectuée par clonage .

Assignation d'un objet

```
<?php
$assigned = $instance;
$reference =& $instance;

$instance->var = '$assigned aura cette valeur';

$instance = null; // $instance et $reference deviennent null

var_dump($instance);
var_dump($reference);
var_dump($assigned);
?>
```

L'exemple ci-dessus va afficher :

```
NULL
NULL
object(SimpleClass)#1 (1) {
  ["var"]=>
  string(30) "$assigned aura cette valeur"
}
```

5.10.2.3 Le mot clé extends

Une classe peut hériter des méthodes et des membres d'une autre classe en utilisant le mot clé `extends` dans la déclaration. Il n'est pas possible d'étendre de multiples classes, une classe peut uniquement hériter d'une seule classe de base.

Les méthodes et membres hérités peuvent être surchargés, à moins que la classe parent ait défini une méthode comme final . Pour surcharger, il suffit de redéclarer la méthode avec le même nom que celui défini dans la classe parent. Il est possible d'accéder à une méthode ou un membre surchargé avec l'opérateur parent:::

Héritage d'une classe simple

```
<?php
// Rappel
class SimpleClass
{
    // déclaration d'un membre
    public $var = 'une valeur par défaut';

    // déclaration de la méthode
    public function displayVar() {
        echo $this->var;
    }
}

// extension de la classe
class ExtendClass extends SimpleClass
{
    // Redéfinition de la méthode parent
```

```
function displayVar()
{
    echo "Classe étendue\n";
    parent::displayVar();
}
}

$extended = new ExtendClass();
$extended->displayVar();
?>
```

L'exemple ci-dessus va afficher :

Classe étendue
une valeur par défaut

5.10.3 Auto-chargement de classes

De nombreux développeurs qui créent des applications orientées objet, créent un fichier source par définition de classe. L'inconvénient majeur de cette méthode est d'avoir à écrire une longue liste d'inclusions de fichier classes au début de chaque script : une inclusion par classe.

En PHP 5, ce n'est plus nécessaire. Vous pouvez définir la fonction `__autoload` qui va automatiquement être appelée si une classe n'est pas encore définie au moment de son utilisation. Grâce à elle, vous avez une dernière chance pour inclure une définition de classe, avant que PHP ne déclare une erreur.

Note

Les exceptions lancées depuis la fonction `__autoload` ne peuvent être interceptées par un bloc `catch` : elles provoqueront une erreur fatale.

Exemple avec `__autoload`

Cet exemple tente de charger les classes `MaClasse1` et `MaClasse2`, dans les fichiers `MaClasse1.php` et `MaClasse2.php` respectivement.

```
<?php
function __autoload($class_name) {
    require_once $class_name . '.php';
}

$obj1 = new MaClasse1();
$obj2 = new MaClasse2();
?>
```

5.10.4 Constructeurs et destructeurs

5.10.4.1 Constructeurs

```
void __construct ( mixed args , ... )
```

PHP 5 permet aux développeurs de déclarer des constructeurs pour les classes. Les classes qui possèdent une méthode constructeur appellent cette méthode à chaque création d'une nouvelle instance de l'objet, ce qui est intéressant pour toutes les initialisations dont l'objet a besoin avant

d'être utilisé.

Note

Les constructeurs parents ne sont pas appelés implicitement si la classe enfant définit un constructeur. Si vous voulez utiliser un constructeur parent, il sera nécessaire de faire appel à `parent::__construct()`.

Exemple d'utilisation des nouveaux constructeurs unifiés

```
<?php
class BaseClass {
    function __construct() {
        print "In BaseClass constructor\n";
    }
}

class SubClass extends BaseClass {
    function __construct() {
        parent::__construct();
        print "In SubClass constructor\n";
    }
}

$obj = new BaseClass();
$obj = new SubClass();
?>
```

Pour des raisons de compatibilité ascendante, si PHP 5 ne peut pas trouver une fonction `__construct()` pour une classe donnée, il cherchera une fonction constructeur représentée, comme dans l'ancien style (PHP < 5), par le nom de la classe. Effectivement, cela signifie que le seul cas où il pourrait y avoir un problème de compatibilité est celui où votre classe contiendrait une méthode nommée `__construct()` et que vous en ayez un autre usage.

5.10.4.2 Destructeurs

`void __destruct ()`

PHP 5 introduit un concept de destructeur similaire aux autres langages orientés objet, comme le C++ . La méthode destructeur doit être appelée aussitôt que toutes les références à un objet particulier sont effacées ou lorsque l'objet est explicitement détruit.

Exemple avec un Destructeur

```
<?php
class MyDestructableClass {
    function __construct() {
        print "In constructor\n";
        $this->name = "MyDestructableClass";
    }

    function __destruct() {
        print "Destruction de " . $this->name . "\n";
    }
}

$obj = new MyDestructableClass();
?>
```

Tout comme le constructeur, le destructeur parent n'est pas appelé implicitement par le moteur. Pour exécuter le destructeur parent, vous devez appeler explicitement la fonction `parent::__destruct` dans le corps du destructeur.

Note

Le destructeur est appelé durant l'arrêt du script, donc les en-têtes sont toujours déjà envoyés.

Note

Tenter de lancer une exception depuis un destructeur résulte en une erreur fatale.

5.10.5 Visibilité

La visibilité d'une propriété ou d'une méthode peut être définie en préfixant la déclaration avec un mot-clé : `public` , `protected` ou `private` . Les éléments déclarés publics (`public`) peuvent être utilisés par n'importe quelle partie du programme. L'accès aux éléments protégés (`protected`) est limité aux classes et parents hérités (et à la classe qui a défini l'élément). L'accès aux éléments privés (`private`) est uniquement réservé à la classe qui les a définis.

5.10.5.1 Visibilité des membres

Les classes membres doivent être définies comme publiques, protégées ou privées.

Déclaration des membres

```
<?php
/**
 * Définition de MyClass
 */
class MyClass
{
    public $public = 'Public';
    protected $protected = 'Protected';
    private $private = 'Private';

    function printHello()
    {
        echo $this->private;
        echo $this->protected;
        echo $this->private;
    }
}

$obj = new MyClass();
echo $obj->public; // Fonctionne
echo $obj->protected; // Erreur fatale
echo $obj->private; // Erreur fatale
$obj->printHello(); // Affiche Public, Protected et Private

/**
 * Définition de MyClass2
 */
class MyClass2 extends MyClass
{
    // On peut redéclarer les éléments publics ou protégés, mais pas les privés
    protected $protected = 'Protected2';

    function printHello()
    {
        echo $this->public;
        echo $this->protected;
        echo $this->private;
    }
}

$obj2 = new MyClass2();
```

```

echo $obj->public; // Fonctionne
echo $obj2->private; // Indéfini
echo $obj2->protected; // Erreur fatale
$obj2->printHello(); // Affiche Public, Protected2 et non Private

?>

```

Note

La méthode de déclaration de variable en PHP 4 avec le mot clé **var** n'est plus valide pour les objets en PHP 5. Pour des raisons de compatibilité, une variable déclarée ainsi aura automatiquement une visibilité publique et une erreur de niveau E_STRICT sera générée.

5.10.5.2 Visibilité des méthodes

Les méthodes des classes doivent être définies en tant que publiques, privées ou protégées. Les méthodes sans déclaration seront automatiquement définies comme étant publiques.

Déclaration d'une méthode

```

<?php
/**
 * Définition de MyClass
 */
class MyClass
{
    // Les constructeurs doivent être publics
    public function __construct() { }

    // Déclaration d'une méthode publique
    public function MyPublic() { }

    // Déclaration d'une méthode protégée
    protected function MyProtected() { }

    // Déclaration d'une méthode privée
    private function MyPrivate() { }

    // Celle-ci sera publique
    function Foo()
    {
        $this->MyPublic();
        $this->MyProtected();
        $this->MyPrivate();
    }
}

$myclass = new MyClass;
$myclass->MyPublic(); // Fonctionne
$myclass->MyProtected(); // Erreur fatale
$myclass->MyPrivate(); // Erreur fatale
$myclass->Foo(); // Public, Protected et Private fonctionnent

/**
 * Définition de MyClass2
 */
class MyClass2 extends MyClass
{
    // Celle-ci sera publique
    function Foo2()
    {
        $this->MyPublic();
        $this->MyProtected();
        $this->MyPrivate(); // Erreur fatale
    }
}

```

```

    }
}

$myclass2 = new MyClass2;
$myclass2->MyPublic(); // Fonctionne
$myclass2->Foo2(); // Public et Protected fonctionnent, non pas Private
?>

```

5.10.6 L'opérateur de résolution de portée (::)

L'opérateur de résolution de portée (aussi appelé Paamayim Nekudotayim) ou, en termes plus simples, le symbole "double deux points" (::), fournit un moyen d'accéder aux membres statiques ou constants ainsi qu'aux éléments redéfinis par la classe.

Lorsque vous référencez ces éléments en dehors de la définition de la classe, utilisez le nom de la classe.

Paamayim Nekudotayim peut sembler un choix étrange pour un double deux points. Cependant, au moment de l'écriture du Zend Engine 0.5 (fournit avec PHP 3), c'est le nom choisi par le groupe Zend. En fait, cela signifie un double deux points... en Hébreu !

:: en dehors de la définition de la classe

```

<?php
class MyClass {
    const CONST_VALUE = 'Une valeur constante';
}

echo MyClass::CONST_VALUE;
?>

```

Deux mots-clé spéciaux, `self` et `parent`, sont utilisés pour accéder aux membres ou aux méthodes depuis la définition de la classe.

:: depuis la définition de la classe

```

<?php
class OtherClass extends MyClass
{
    public static $my_static = 'variable statique';

    public static function doubleColon() {
        echo parent::CONST_VALUE . "\n";
        echo self::$my_static . "\n";
    }
}

OtherClass::doubleColon();
?>

```

Lorsqu'une classe étendue redéfinit une méthode de la classe parente, PHP n'appellera pas la méthode d'origine. Il appartient à la méthode dérivée d'appeler la méthode d'origine en cas de besoin. Cela est également valable pour les définitions des constructeurs et destructeurs, les surcharges et les méthodes magiques.

Appel d'une méthode parent

```

<?php
class MyClass
{
    protected function myFunc() {

```

```

        echo "MyClass::myFunc()\n";
    }
}

class OtherClass extends MyClass
{
    // Dépassement de la définition parent
    public function myFunc() {

        // Mais appel de la fonction parent
        parent::myFunc();
        echo "OtherClass::myFunc()\n";
    }
}

$class = new OtherClass();
$class->myFunc();
?>

```

5.10.7 Statique

Le fait de déclarer des membres ou des méthodes comme statiques vous permet d'y accéder sans avoir besoin d'instancier la classe. Un membre déclaré comme statique ne peut être accédé avec l'objet instancié d'une classe (bien qu'une méthode statique le peut).

La déclaration `static` doit être faite après la déclaration de visibilité. Pour des raisons de compatibilité avec PHP 4, si aucune déclaration de visibilité n'est utilisée, alors le membre ou la méthode sera traité comme s'il avait été déclaré comme `public`.

Comme les méthodes statiques sont appelables sans instance d'objet créée, la pseudo variable `$this` n'est pas disponible dans la méthode déclarée en tant que statique.

En fait, les appels de méthodes statiques sont résolus au moment de la compilation. Lorsque l'on utilise un nom de classe explicite, la méthode est déjà identifiée complètement et aucune notion d'héritage n'est appliquée. Si l'appel est effectuée par le mot clé `self`, alors `self` est traduit en la classe courante, qui est la classe appartenant au code. Ici aussi, aucune notion d'héritage n'est appliquée.

On ne peut pas accéder à des propriétés statiques à travers l'objet en utilisant l'opérateur `->`.

L'appel non-statique à des méthodes statiques génère une alerte de degré `E_STRICT`.

Exemple avec un membre statique

```

<?php
class Foo
{
    public static $my_static = 'foo';

    public function staticValue() {
        return self::$my_static;
    }
}

class Bar extends Foo
{
    public function fooStatic() {
        return parent::$my_static;
    }
}

```

```

}

print Foo::$my_static . "\n";

$foo = new Foo();
print $foo->staticValue() . "\n";
print $foo->my_static . "\n";      // propriété my_static non définie

// $foo::my_static n'est pas possible

print Bar::$my_static . "\n";
$bar = new Bar();
print $bar->fooStatic() . "\n";
?>

```

Exemple avec une méthode statique

```

<?php
class Foo
{
    public static function aStaticMethod() {
        // ...
    }
}

Foo::aStaticMethod();
?>

```

5.10.8 Constantes de classe

Il est possible de définir des valeurs constantes à l'intérieur d'une classe, qui ne seront pas modifiables. Les constantes diffèrent des variables normales du fait qu'on n'utilise pas le symbole \$ pour les déclarer ou les utiliser. Tout comme pour les membres statiques, on ne peut pas accéder aux valeurs constantes depuis une instance de l'objet (en utilisant \$object::constant).

La valeur doit être une expression constante, non (par exemple) une variable, un membre de la classe, le résultat d'une opération mathématique ou un appel de fonction.

Définition et utilisation d'une constante de classe

```

<?php
class MyClass
{
    const constant = 'valeur constante';

    function showConstant() {
        echo self::constant . "\n";
    }
}

echo MyClass::constant . "\n";

$class = new MyClass();
$class->showConstant();
// echo $class::constant; n'est pas autorisé
?>

```

5.10.9 Abstraction de classes

PHP 5 introduit les classes et les méthodes abstraites. Il n'est pas autorisé de créer une instance d'une classe définie comme abstraite. Toutes les classes contenant au moins une méthode abstraite doivent également être abstraites. Pour définir une méthode abstraite, il faut simplement déclarer la signature de la méthode et ne fournir aucune implémentation.

Lors de l'héritage depuis une classe abstraite, toutes les méthodes marquées comme abstraites dans la déclaration de la classe parent doivent être définies par l'enfant ; de plus, ces méthodes doivent être définies avec la même (ou plus faible) visibilité . Par exemple, si la méthode abstraite est définie comme protégée, l'implémentation de la fonction doit être définie en tant que protégée ou publique.

Exemple de classe abstraite

```
<?php
abstract class AbstractClass
{
    // Force la classe étendue à définir cette méthode
    abstract protected function getValue();
    abstract protected function prefixValue($prefix);

    // méthode commune
    public function printOut() {
        print $this->getValue() . "\n";
    }
}

class ConcreteClass1 extends AbstractClass
{
    protected function getValue() {
        return "ConcreteClass1";
    }

    public function prefixValue($prefix) {
        return "{$prefix}ConcreteClass1";
    }
}

class ConcreteClass2 extends AbstractClass
{
    public function getValue() {
        return "ConcreteClass2";
    }

    public function prefixValue($prefix) {
        return "{$prefix}ConcreteClass2";
    }
}

$class1 = new ConcreteClass1;
$class1->printOut();
echo $class1->prefixValue('FOO_') . "\n";

$class2 = new ConcreteClass2;
$class2->printOut();
echo $class2->prefixValue('FOO_') . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
ConcreteClass1
FOO_ConcreteClass1
ConcreteClass2
FOO_ConcreteClass2
```

Du code ancien n'ayant aucune classe ou fonction nommée abstract devrait fonctionner sans modifications.

5.10.10 Interfaces

Les interfaces objet vous permettent de créer du code qui spécifie quelles méthodes et variables une classe peut implémenter, sans avoir à définir comment ces méthodes seront gérées.

Les interfaces sont définies en utilisant le mot clé `interface`, de la même façon qu'une classe standard mais sans aucun contenu de méthode.

Toutes les méthodes déclarées dans une interface doivent être publiques.

`implements`

Pour implémenter une interface, l'opérateur `implements` est utilisé. Toutes les méthodes de l'interface doivent être implémentées dans une classe ; si ce n'est pas le cas, une erreur fatale sera émise. Les classes peuvent implémenter plus d'une interface en séparant chaque interface par une virgule.

Exemples

Exemple d'interface

```
<?php
// Declaration de l'interface 'iTemplate'
interface iTemplate
{
    public function setVariable($name, $var);
    public function getHtml($template);
}

// Implémentation de l'interface
// Ceci va fonctionner
class Template implements iTemplate
{
    private $vars = array();

    public function setVariable($name, $var)
    {
        $this->vars[$name] = $var;
    }

    public function getHtml($template)
    {
        foreach($this->vars as $name => $value) {
            $template = str_replace('{ ' . $name . ' }', $value, $template);
        }

        return $template;
    }
}

// Ceci ne fonctionnera pas
```

```
// Fatal error: Class BadTemplate contains 1 abstract methods
// and must therefore be declared abstract (iTemplate::getHtml)
class BadTemplate implements iTemplate
{
    private $vars = array();

    public function setVariable($name, $var)
    {
        $this->vars[$name] = $var;
    }
}
?>
```

Voir aussi l'opérateur [instanceof](#) .

5.10.11 Surcharge

Les appels de méthodes et l'accès aux membres peuvent être surchargés via les méthodes `__call` , `__get` et `__set` . Ces méthodes ne seront déclenchées que si votre objet, hérité ou non, ne contient pas le membre ou la méthode auquel vous tentez d'accéder. Toutes les méthodes surchargées doivent être définies en tant que `public` .

Depuis PHP 5.1.0, il est également possible de surcharger les fonctions `isset` et `unset` via, respectivement, les méthodes `__isset` et `__unset`.

5.10.11.1 Surcharge de membre

```
void __set ( string name , mixed value ) mixed __get ( string name ) bool __isset ( string name ) void __unset ( string name )
```

Les membres d'une classe peuvent être surchargés afin d'exécuter un code spécial contenu dans vos méthodes `__set` et `__get` . Le paramètre `$name` est le nom de la variable qui doit être définie ou recherchée. Le paramètre `$valeur` de la méthode `__set` spécifie la nouvelle valeur à donner à la variable `$name` .

Exemple de surcharge avec `__get`, `__set`, `__isset` et `__unset`

```
<?php
class Setter
{
    public $n;
    private $x = array("a" => 1, "b" => 2, "c" => 3);

    private function __get($nm)
    {
        echo "Récupération de [$nm]\n";

        if (isset($this->x[$nm])) {
            $r = $this->x[$nm];
            echo "Retour : $r\n";
            return $r;
        } else {
            echo "Rien!\n";
        }
    }

    private function __set($nm, $val)
    {
        echo "Définition de [$nm] à $val\n";
    }
}
```

```

    if (isset($this->x[$nm])) {
        $this->x[$nm] = $val;
        echo "OK!\n";
    } else {
        echo "Pas OK!\n";
    }
}

private function __isset($nm)
{
    echo "Vérifie si $nm est défini\n";

    return isset($this->x[$nm]);
}

private function __unset($nm)
{
    echo "Libération de $nm\n";

    unset($this->x[$nm]);
}
}

$foo = new Setter();
$foo->n = 1;
$foo->a = 100;
$foo->a++;
$foo->z++;
var_dump($foo);

var_dump(isset($foo->a)); //true
unset($foo->a);
var_dump(isset($foo->a)); //false

// ceci ne passera pas via la méthode __isset()
// parce que 'n' est une propriété publique
var_dump(isset($foo->n));

var_dump($foo);
?>

```

L'exemple ci-dessus va afficher :

```

Définition de [a] à 100
OK!
Récupération de [a]
Retour : 100
Définition de [a] à 101
OK!
Récupération de [z]
Rien!
Définition de [z] à 1
Pas OK!

Vérifie si a est défini
bool(true)
Libération de a
Vérifie si a est défini
bool(false)

```

```

bool(true)

object(Setter)#1 (2) {
  ["n"]=>
  int(1)
  ["x:private"]=>
  array(2) {
    ["b"]=>
    int(2)
    ["c"]=>
    int(3)
  }
}

```

5.10.11.2 Surcharge de méthode

mixed `__call` (string name , array arguments)

Les méthodes d'une classe peuvent être surchargées pour exécuter un code spécial contenu dans votre méthode `__call`. Le paramètre \$name utilisé est le nom de la fonction que l'on souhaite utiliser. Les arguments qui étaient passés à la fonction sont récupérés dans un tableau contenu dans le paramètre \$arguments. La valeur retournée par la méthode `__call` sera celle retournée au point d'appel de la méthode d'origine.

Exemple de surcharge avec `__call`

```

<?php
class Caller
{
    private $x = array(1, 2, 3);

    public function __call($m, $a)
    {
        print "Méthode $m appelée :\n";
        var_dump($a);
        return $this->x;
    }
}

$foo = new Caller();
$a = $foo->test(1, "2", 3.4, true);
var_dump($a);
?>

```

L'exemple ci-dessus va afficher :

Méthode test appelée :

```

array(4) {
  [0]=>
  int(1)
  [1]=>
  string(1) "2"
  [2]=>
  float(3.4)
  [3]=>
  bool(true)
}

```

```

}
array(3) {
[0]=>
int(1)
[1]=>
int(2)
[2]=>
int(3)
}

```

5.10.12 Parcours d'objets

PHP 5 fournit une façon de définir les objets de manière à ce qu'on puisse parcourir une liste de membres avec une structure `foreach`. Par défaut, toutes les propriétés visibles seront utilisées pour le parcours.

Parcours d'objet simple

```

<?php
class MyClass
{
    public $var1 = 'valeur 1';
    public $var2 = 'valeur 2';
    public $var3 = 'valeur 3';

    protected $protected = 'variable protégée';
    private $private = 'variable privée';

    function iterateVisible() {
        echo "MyClass::iterateVisible:\n";
        foreach($this as $key => $value) {
            print "$key => $value\n";
        }
    }
}

$class = new MyClass();

foreach($class as $key => $value) {
    print "$key => $value\n";
}
echo "\n";

$class->iterateVisible();

```

L'exemple ci-dessus va afficher :

```

var1 => valeur 1
var2 => valeur 2
var3 => valeur 3

```

```

MyClass::iterateVisible:
var1 => valeur 1
var2 => valeur 2
var3 => valeur 3
protected => variable protégée
private => variable privée

```

Comme nous le montre l'affichage, l'itération `foreach` affiche toutes les variables visibles disponibles. Pour aller plus loin, vous pouvez implémenter l' interface interne de PHP 5 nommée `Iterator` . Ceci permet de déterminer comment l'objet doit être parcouru.

Itération d'un objet implémentant un itérateur

```
<?php
class MyIterator implements Iterator
{
    private $var = array();

    public function __construct($array)
    {
        if (is_array($array) ) {
            $this->var = $array;
        }
    }

    public function rewind() {
        echo "rembobinage\n";
        reset($this->var);
    }

    public function current() {
        $var = current($this->var);
        echo "actuel : $var\n";
        return $var;
    }

    public function key() {
        $var = key($this->var);
        echo "clé : $var\n";
        return $var;
    }

    public function next() {
        $var = next($this->var);
        echo "suivant : $var\n";
        return $var;
    }

    public function valid() {
        $var = $this->current() !== false;
        echo "valide : {$var}\n";
        return $var;
    }
}

$values = array(1,2,3);
$it = new MyIterator($values);

foreach ($it as $a => $b) {
    print "$a: $b\n";
}
```

L'exemple ci-dessus va afficher :

```
rembobinage
actuel : 1
valide : 1
actuel : 1
clé : 0
```

```

0: 1
suivant : 2
actuel : 2
valide : 1
actuel : 2
clé : 1
1: 2
suivant : 3
actuel : 3
valide : 1
actuel : 3
clé : 2
2: 3
suivant :
actuel :
valide :

```

Vous pouvez également définir votre classe de façon à ce qu'elle n'ait pas besoin de définir toutes les fonctions Iterator en implémentant simplement l'interface PHP 5 IteratorAggregate .

Itération d'un objet implémentant IteratorAggregate

```

<?php
class MyCollection implements IteratorAggregate
{
    private $items = array();
    private $count = 0;

    // Définition requise de l'interface IteratorAggregate
    public function getIterator() {
        return new MyIterator($this->items);
    }

    public function add($value) {
        $this->items[$this->count++] = $value;
    }
}

$coll = new MyCollection();
$coll->add('valeur 1');
$coll->add('valeur 2');
$coll->add('valeur 3');

foreach ($coll as $key => $val) {
    echo "clé/valeur : [$key -> $val]\n\n";
}
?>

```

L'exemple ci-dessus va afficher :

```

rembobinage
actuel : valeur 1
valide : 1
actuel : valeur 1
clé : 0
clé/valeur : [0 -> valeur 1]

suivant : valeur 2
actuel : valeur 2

```

```
valide : 1
actuel : valeur 2
clé : 1
clé/valeur : [1 -> valeur 2]
```

```
suisvant : valeur 3
actuel : valeur 3
valide : 1
actuel : valeur 3
clé : 2
clé/valeur : [2 -> valeur 3]
```

```
suisvant :
actuel :
valide :
```

Note

Pour plus d'exemples sur le parcours d'objets, lisez la section sur l' [extension SPL](#) .

5.10.13 Masques

Les masques sont un moyen de décrire les meilleures pratiques et les bonnes conceptions. Ils proposent une solution flexible aux problèmes habituels de programmation.

5.10.13.1 Usine

Le masque d'usine permet l'instanciation d'objets durant l'exécution. Il est appelé "masque d'usine" puisqu'il est responsable de la "fabrication" d'un objet. Un paramètre d'usine reçoit le nom de la classe pour l'instancier en tant qu'argument.

Méthode de paramètre d'usine

```
<?php
class Exemple
{
    // La méthode de paramètre d'usine
    public static function factory($type)
    {
        if (include_once 'Drivers/' . $type . '.php') {
            $classname = 'Driver_' . $type;
            return new $classname;
        } else {
            throw new Exception ('Driver non trouvé');
        }
    }
}
?>
```

Définir cette méthode dans une classe permet de charger un pilote à la volée. Si la classe Exemple était une classe d'abstraction de base de données, le chargement des pilotes MySQL et SQLite pourrait être effectué comme ceci :

```
<?php
// Chargement du driver MySQL
$mysql = Exemple::factory('MySQL');

// Chargement du driver SQLite
```

```
$sqlite = Example::factory('SQLite');
?>
```

5.10.13.2 Singleton

Le masque singleton est utilisé dans les situations où l'on a besoin qu'il n'y ait une unique instance d'une certaine classe. L'exemple le plus commun est une connexion à une base de données. L'implémentation de ce masque permet au développeur de rendre cette unique instance facilement accessible par beaucoup d'autres objets.

Fonction Singleton

```
<?php
class Example
{
    // instance de la classe
    private static $instance;

    // Un constructeur privé ; empêche la création directe d'objet
    private function __construct()
    {
        echo 'Je suis construit';
    }

    // La méthode singleton
    public static function singleton()
    {
        if (!isset(self::$instance)) {
            $c = __CLASS__;
            self::$instance = new $c;
        }

        return self::$instance;
    }

    // Exemple d'une méthode
    public function bark()
    {
        echo 'Woof!';
    }

    // Préviens les utilisateurs sur le clonage de l'instance
    public function __clone()
    {
        trigger_error('Le clonage n\'est pas autorisé.', E_USER_ERROR);
    }
}
?>
```

Ceci autorise une unique instance de la classe Example .

```
<?php
// Ceci échoue car le constructeur est privé
$test = new Example;

// Ceci récupère toujours une seule instance de la classe
$test = Example::singleton();
$test->bark();

// Ceci provoque une erreur E_USER_ERROR.
$test_clone = clone($test);
```

?>

5.10.14 Méthodes magiques

Les noms de fonction `__construct`, `__destruct` (voir les [Constructeurs et Destructeurs](#)), `__call`, `__get`, `__set`, `__isset`, `__unset` (voir la [surcharge](#)), `__sleep`, `__wakeup`, `__toString`, `__set_state`, `__clone` et `__autoload` sont magiques dans les classes PHP. Vous ne pouvez pas utiliser ces noms de fonction dans aucune de vos classes sauf si vous voulez modifier le comportement associé à ces fonctions magiques.

Attention

PHP réserve tous les noms de fonctions commençant par `__` pour les fonctions magiques. Il est recommandé de ne pas utiliser de noms de fonctions commençant par `__` sauf si vous voulez des fonctionnalités magiques documentées.

5.10.14.1 `__sleep` et `__wakeup`

La fonction [serialize](#) vérifie si votre classe a une fonction avec le nom magique `__sleep`. Si c'est le cas, cette fonction sera exécutée avant toute linéarisation. Elle peut nettoyer l'objet et elle est supposée retourner un tableau avec les noms de toutes les variables de l'objet qui doivent être linéarisés.

Le but avoué de `__sleep` est de clore toutes les connexions aux bases de données que l'objet peut avoir, valider les données en attente ou effectuer des tâches de nettoyage. De plus, cette fonction est utile si vous avez de très gros objets qui n'ont pas besoin d'être sauvegardés en totalité.

Réciproquement, la fonction [unserialize](#) vérifie la présence d'une fonction dont le nom est le nom magique `__wakeup`. Si elle est présente, cette fonction peut reconstruire toute ressource que l'objet possède.

Le but avoué de `__wakeup` est de rétablir toute connexion base de données qui aurait été perdue durant la linéarisation et d'effectuer des tâches de réinitialisation.

Utilisation de Sleep et de wakeup

```
<?php
class Connection {
    protected $link;
    private $server, $username, $password, $db;

    public function __construct($server, $username, $password, $db)
    {
        $this->server = $server;
        $this->username = $username;
        $this->password = $password;
        $this->db = $db;
        $this->connect();
    }

    private function connect()
    {
        $this->link = mysql_connect($this->server, $this->username, $this->password);
        mysql_select_db($this->db, $this->link);
    }

    public function __sleep()
    {
```

```

        mysql_close($this->link);
    }

    public function __wakeup()
    {
        $this->connect();
    }
}
?>

```

5.10.14.2 __toString

La méthode `__toString` détermine comment la classe doit réagir lorsqu'elle est convertie en chaîne de caractères.

Exemple simple

```

<?php
// Déclaration d'une classe simple
class ClasseTest
{
    public $foo;

    public function __construct($foo) {
        $this->foo = $foo;
    }

    public function __toString() {
        return $this->foo;
    }
}

$class = new ClasseTest('Bonjour');
echo $class;
?>

```

L'exemple ci-dessus va afficher :

Bonjour

Il est important de noter que la méthode `__toString` ne sera appelée que si elle est directement combinée avec `echo` ou `print`.

Cas où `__toString` est appelée

```

<?php
// __toString appelée
echo $class;

// __toString appelée (reste un paramètre normal pour echo)
echo 'text', $class;

// __toString non appelée (opérateur de concaténation utilisé en premier)
echo 'text' . $class;

// __toString non appelée (conversion en une chaîne de caractères en premier)
echo (string) $class;

// __toString non appelée (conversion en une chaîne de caractères en premier)
echo "text $class";
?>

```

5.10.14.3 __set_state

Cette méthode statique est appelée pour les classes exportées par la fonction var_export depuis PHP 5.1.0.

Le seul paramètre de cette méthode est un tableau contenant les propriétés exportées sous la forme `array('propriété' => valeur, ...)`.

5.10.15 Mot clé "final"

PHP 5 introduit le mot-clé " final " qui empêche les classes filles de surcharger une méthode en en préfixant la définition par le mot-clé " final ". Si la classe elle-même est définie comme finale, elle ne pourra pas être étendue.

Exemple de méthode finale

```
<?php
class BaseClass {
    public function test() {
        echo "BaseClass::test() appelé\n";
    }

    final public function moreTesting() {
        echo "BaseClass::moreTesting() appelé\n";
    }
}

class ChildClass extends BaseClass {
    public function moreTesting() {
        echo "ChildClass::moreTesting() appelé\n";
    }
}
// Résultat : Fatal error: Cannot override final method BaseClass::moreTesting()
?>
```

Exemple de classe finale

```
<?php
final class BaseClass {
    public function test() {
        echo "BaseClass::test() appelé\n";
    }

    // Ici, peut importe si vous spécifié la fonction en final ou pas
    final public function moreTesting() {
        echo "BaseClass::moreTesting() appelé\n";
    }
}

class ChildClass extends BaseClass {
}
// Résultat : Fatal error: Class ChildClass may not inherit from final class (BaseClass)
?>
```

5.10.16 Clonage d'objets

Le fait de créer une copie d'un objet possédant exactement les mêmes propriétés n'est pas toujours le comportement que l'on souhaite. Un bon exemple pour illustrer le besoin d'un constructeur de copie : si vous avez un objet qui représente une fenêtre GTK et que l'objet contient la ressource représentant cette fenêtre GTK, lorsque vous créez une copie vous pouvez vouloir créer une

nouvelle fenêtre avec les mêmes propriétés mais que le nouvel objet contienne une ressource représentant la nouvelle fenêtre.

Un autre exemple : si votre objet `$a` contient une référence vers un autre objet `$b` qui est en cours d'utilisation, lorsque vous copiez l'objet parent (`$a`), vous voulez créer une nouvelle instance de `$b` pour que votre copie possède sa propre instance.

Une copie d'objet est créée en utilisant le mot clé `clone` (qui fait appel à la méthode `__clone` de l'objet, si elle a été définie). La méthode `__clone` d'un objet ne peut être appelée directement.

```
<?php
$copie_d_objet = clone $objet;
?>
```

Lorsqu'un objet est cloné, PHP 5 effectue une copie superficielle de toutes les propriétés de l'objet. Toutes les propriétés qui sont des références à d'autres variables demeureront des références. Si une méthode `__clone()` est définie, la méthode `__clone()` du nouvel objet sera appelée pour permettre à chaque propriété qui doit l'être d'être modifiée.

Exemple de duplication d'objets

```
<?php
class SubObject
{
    static $instances = 0;
    public $instance;

    public function __construct() {
        $this->instance = ++self::$instances;
    }

    public function __clone() {
        $this->instance = ++self::$instances;
    }
}

class MyCloneable
{
    public $objet1;
    public $objet2;

    function __clone()
    {
        // Force la copie de this->object, sinon
        // il pointera vers le même objet.
        $this->objet1 = clone($this->objet1);
    }
}

$objj = new MyCloneable();

$objj->objet1 = new SubObject();
$objj->objet2 = new SubObject();

$objj2 = clone $objj;

print("Objet original :\n");
print_r($objj);

print("Objet cloné :\n");
print_r($objj2);
```

```
?>
```

L'exemple ci-dessus va afficher :

```
Object original :
MyCloneable Object
(
[object1] => SubObject Object
(
[instance] => 1
)
)
[object2] => SubObject Object
(
[instance] => 2
)
)
Object cloné :
MyCloneable Object
(
[object1] => SubObject Object
(
[instance] => 3
)
)
[object2] => SubObject Object
(
[instance] => 2
)
)
)
```

5.10.17 Comparaison d'objets

En PHP 5, la comparaison d'objets est plus compliquée qu'en PHP 4 afin d'être plus proche du comportement des langages orientés objet (bien que PHP n'en soit pas un).

Lors de l'utilisation de l'opérateur de comparaison `==`, les objets sont comparés de manière simple, à savoir : deux objets sont égaux s'ils ont les mêmes attributs et valeurs, et qu'ils sont des instances de la même classe.

D'un autre côté, lors de l'utilisation de l'opérateur d'identité (`===`), les objets sont identiques uniquement s'ils font référence à la même instance de la même classe.

Un exemple va illustrer ces règles.

Exemple de comparaison d'objets en PHP 5

```
<?php
function bool2str($bool)
{
```

```

    if ($bool === false) {
        return 'FALSE';
    } else {
        return 'TRUE';
    }
}

function compareObjects(&$o1, &$o2)
{
    echo 'o1 == o2 : '.bool2str($o1 == $o2)."\n";
    echo 'o1 != o2 : '.bool2str($o1 != $o2)."\n";
    echo 'o1 === o2 : '.bool2str($o1 === $o2)."\n";
    echo 'o1 !== o2 : '.bool2str($o1 !== $o2)."\n";
}

class Flag
{
    public $flag;

    function Flag($flag = true) {
        $this->flag = $flag;
    }
}

class OtherFlag
{
    public $flag;

    function OtherFlag($flag = true) {
        $this->flag = $flag;
    }
}

$o = new Flag();
$p = new Flag();
$q = $o;
$r = new OtherFlag();

echo "Deux instances de la même classe\n";
compareObjects($o, $p);

echo "\nDeux références sur le même objet\n";
compareObjects($o, $q);

echo "\nInstances de classes différentes\n";
compareObjects($o, $r);
?>

```

L'exemple ci-dessus va afficher :

Deux instances de la même classe

```

o1 == o2 : TRUE
o1 != o2 : FALSE
o1 === o2 : FALSE
o1 !== o2 : TRUE

```

Deux références sur le même objet

```

o1 == o2 : TRUE
o1 != o2 : FALSE
o1 === o2 : TRUE
o1 !== o2 : FALSE

```

Instances de classes différentes

`o1 == o2` : FALSE

`o1 != o2` : TRUE

`o1 === o2` : FALSE

`o1 !== o2` : TRUE

5.10.18 Réflexion

5.10.18.1 Introduction

PHP 5 introduit API de réflexion complète qui permet de faire du reverse-engineering sur les classes, les interfaces, les fonctions et les méthodes tout comme les extensions. L'API de réflexion permet également d'obtenir les commentaires de la documentation pour les fonctions, les classes et les méthodes.

L'API de réflexion est une extension orientée objet du Moteur Zend, constituée des classes suivantes :

API de réflexion

```
<?php
class Reflection { }
interface Reflector { }
class ReflectionException extends Exception { }
class ReflectionFunction implements Reflector { }
class ReflectionParameter implements Reflector { }
class ReflectionMethod extends ReflectionFunction { }
class ReflectionClass implements Reflector { }
class ReflectionObject extends ReflectionClass { }
class ReflectionProperty implements Reflector { }
class ReflectionExtension implements Reflector { }
?>
```

Note

Pour plus de détails sur ces classes, lisez les chapitres suivants.

Si nous exécutons le code de l'exemple ci-dessous :

Utilisation basique de l'API de réflexion

```
<?php
Reflection::export(new ReflectionClass('Exception'));
?>
```

L'exemple ci-dessus va afficher :

```
Class [ <internal> class Exception ] {
```

```
- Constants [0] {
}
```

```
- Static properties [0] {
}
```

```
- Static methods [0] {
```

```

}
- Properties [6] {
Property [ <default> protected $message ]
Property [ <default> private $string ]
Property [ <default> protected $code ]
Property [ <default> protected $file ]
Property [ <default> protected $line ]
Property [ <default> private $trace ]
}
- Methods [9] {
Method [ <internal> final private method __clone ] {
}
Method [ <internal> <ctor> public method __construct ] {
}
- Parameters [2] {
Parameter #0 [ <required> $message ]
Parameter #1 [ <required> $code ]
}
Method [ <internal> final public method getMessage ] {
}
Method [ <internal> final public method getCode ] {
}
Method [ <internal> final public method getFile ] {
}
Method [ <internal> final public method getLine ] {
}
Method [ <internal> final public method getTrace ] {
}
Method [ <internal> final public method getTraceAsString ] {
}
Method [ <internal> public method __toString ] {
}
}
}
}

```

5.10.18.2 ReflectionException

ReflectionException étend le standard [Exception](#) et est lancé par l'API Reflection. Aucune méthode spécifique ni de propriété ne sont introduites.

5.10.18.3 ReflectionFunction

La classe ReflectionFunction vous permet de faire du reverse-engineering sur les fonctions.

La classe ReflectionFunction

```
<?php
class ReflectionFunction implements Reflector
{
    final private __clone()
    public object __construct(string name)
    public string __toString()
    public static string export()
    public string getName()
    public bool isInternal()
    public bool isUserDefined()
    public string getFileName()
    public int getStartLine()
    public int getEndLine()
    public string getDocComment()
    public array getStaticVariables()
    public mixed invoke(mixed* args)
    public mixed invokeArgs(array args)
    public bool returnsReference()
    public ReflectionParameter[] getParameters()
    public int getNumberOfParameters()
    public int getNumberOfRequiredParameters()
}
?>
```

Note

`getNumberOfParameters` et `getNumberOfRequiredParameters` ont été ajoutés en PHP 5.0.3, tandis que `invokeArgs` a été ajouté en PHP 5.1.0.

Pour connaître le fonctionnement d'une fonction, vous devez tout d'abord créer une instance de la classe ReflectionFunction . Ainsi, vous pouvez appeler n'importe quelle méthode de cette instance.

Utilisation de la classe ReflectionFunction

```
<?php
/**
 * Un simple compteur
 *
 * @return int
 */
function counter()
{
    static $c = 0;

    return $c++;
}

// Création d'une instance de la classe Reflection_Function
$func = new ReflectionFunction('counter');

// Affichage d'informations basiques
printf(
    "===> The %s function '%s'\n".
    "    declared in %s\n".
    "    lines %d to %d\n",
    $func->isInternal() ? 'internal' : 'user-defined',
    $func->getName(),
    $func->getFileName(),
    $func->getStartLine(),
    $func->getEndline()
);
```

```
// Affichage du commentaire de la documentation
printf("----> Documentation:\n %s\n", var_export($func->getDocComment(), 1));

// Affichage des variables statiques si elles existent
if ($statics = $func->getStaticVariables())
{
    printf("----> Variables statiques : %s\n", var_export($statics, 1));
}

// Appel de la fonction
printf("----> Invocation des résultats dans : ");
var_dump($func->invoke());

// vous pouvez préférer utiliser la méthode export()
echo "\nRésultat de ReflectionFunction::export() :\n";
echo ReflectionFunction::export('counter');
?>
```

Note

La méthode `invoke` accepte un nombre variable d'arguments, tout comme la fonction `call_user_func`.

5.10.18.4 ReflectionParameter

La classe `ReflectionParameter` récupère les informations concernant les paramètres des fonctions ou des méthodes.

La classe ReflectionParameter

```
<?php
class ReflectionParameter implements Reflector
{
    final private __clone()
    public object __construct(string name)
    public string __toString()
    public static string export()
    public string getName()
    public bool isPassedByReference()
    public ReflectionClass getClass()
    public bool isArray()
    public bool allowsNull()
    public bool isOptional()
    public bool isDefaultValueAvailable()
    public mixed getDefaultValue()
}
?>
```

Note

`getDefaultValue`, `isDefaultValueAvailable` et `isOptional` ont été ajoutés en PHP 5.0.3, tandis que `isArray` a été ajouté en PHP 5.1.0.

Pour connaître le fonctionnement des paramètres d'une fonction, vous devez tout d'abord créer une instance de la classe `ReflectionFunction` ou `ReflectionMethod` et ainsi, utiliser leurs méthodes `getParameters` pour récupérer un tableau de paramètres.

Utilisation de la classe ReflectionParameter

```
<?php
function foo($a, $b, $c) { }
function bar(Exception $a, &$b, $c) { }
function baz(ReflectionFunction $a, $b = 1, $c = null) { }
function abc() { }
```

```
// Création d'une instance de la classe Reflection_Function avec le
// paramètre fourni en ligne de commande.
$reflect = new ReflectionFunction($argv[1]);

echo $reflect;

foreach ($reflect->getParameters() as $i => $param) {
    printf(
        "-- Paramètre #%d : %s {\n".
        "  Classe : %s\n".
        "  Autorise NULL : %s\n".
        "  Passé par référence : %s\n".
        "  Est optionnel ? : %s\n".
        "}\n",
        $i,
        $param->getName(),
        var_export($param->getClass(), 1),
        var_export($param->allowsNull(), 1),
        var_export($param->isPassedByReference(), 1),
        $param->isOptional() ? 'oui' : 'non'
    );
}
?>
```

5.10.18.5 ReflectionClass

La classe ReflectionClass vous permet de faire du reverse-engineering sur des classes.

La classe ReflectionClass

```
<?php
class ReflectionClass implements Reflector
{
    final private __clone()
    public object __construct(string name)
    public string __toString()
    public static string export()
    public string getName()
    public bool isInternal()
    public bool isUserDefined()
    public bool isInstantiable()
    public bool hasConstant(string name)
    public bool hasMethod(string name)
    public bool hasProperty(string name)
    public string getFileName()
    public int getStartLine()
    public int getEndLine()
    public string getDocComment()
    public ReflectionMethod getConstructor()
    public ReflectionMethod getMethod(string name)
    public ReflectionMethod[] getMethods()
    public ReflectionProperty getProperty(string name)
    public ReflectionProperty[] getProperties()
    public array getConstants()
    public mixed getConstant(string name)
    public ReflectionClass[] getInterfaces()
    public bool isInterface()
    public bool isAbstract()
    public bool isFinal()
    public int getModifiers()
    public bool isInstance(stdclass object)
    public stdclass newInstance(mixed* args)
    public ReflectionClass getParentClass()
    public bool isSubclassOf(ReflectionClass class)
```

```

public array getStaticProperties()
public mixed getStaticPropertyValue(string name [, mixed default])
public void setStaticPropertyValue(string name, mixed value)
public array getDefaultProperties()
public bool isIterateable()
public bool implementsInterface(string name)
public ReflectionExtension getExtension()
public string getExtensionName()
}
?>

```

Note

hasConstant , hasMethod , hasProperty , getStaticPropertyValue et setStaticPropertyValue ont été ajoutées en PHP 5.1.0.

Pour connaître le fonctionnement d'une classe, vous devez d'abord créer une instance de la classe ReflectionClass . Vous pourrez donc appeler n'importe quelle méthode sur cette instance.

Utilisation de la classe ReflectionClass

```

<?php
interface Linearisable
{
    // ...
}

class Object
{
    // ...
}

/**
 * Une classe compteur
 *
 */
class Compteur extends Object implements Linearisable
{
    const START = 0;
    private static $c = Compteur::START;

    /**
     * Invocation du compteur
     *
     * @access public
     * @return int
     */
    public function count()
    {
        return self::$c++;
    }
}

// Création d'une instance de la classe ReflectionClass
$class = new ReflectionClass('Compteur');

// Affichage d'informations basiques
printf(
    "===> La %s%s %s '%s' [extension de %s]\n".
    "    déclarée dans %s\n".
    "    lignes %d à %d\n".
    "    a le modificateur %d [%s]\n",
    $class->isInternal() ? 'internal' : 'user-defined',
    $class->isAbstract() ? ' abstract' : '',
    $class->isFinal() ? ' final' : '',
    $class->isInterface() ? 'interface' : 'class',
    $class->getName(),

```

```

var_export($class->getParentClass(), 1),
$class->getFileName(),
$class->getStartLine(),
$class->getEndline(),
$class->getModifiers(),
implode(' ', Reflection::getModifierNames($class->getModifiers()))
);

// Affichage du commentaire de la documentation
printf("----> Documentation:\n %s\n", var_export($class->getDocComment(), 1));

// Affichage de l'interface qui implémente cette classe
printf("----> Implémenté :\n %s\n", var_export($class->getInterfaces(), 1));

// Affichage des constantes de la classe
printf("----> Constantes : %s\n", var_export($class->getConstants(), 1));

// Affichage des propriétés de la classe
printf("----> Properties: %s\n", var_export($class->getProperties(), 1));

// Affichage des méthodes de la classe
printf("----> Méthodes : %s\n", var_export($class->getMethods(), 1));

// Si cette classe est instanciable, création d'une instance
if ($class->isInstantiable()) {
    $counter = $class->newInstance();

    echo '----> $counter est uneinstance ? ';
    echo $class->isInstance($counter) ? 'oui' : 'non';

    echo "\n----> Le nouvel objet Object() est une instance ? ";
    echo $class->isInstance(new Object()) ? 'oui' : 'non';
}
?>

```

Note

La méthode `newInstance` accepte un nombre variable d'arguments, tout comme la fonction `call_user_func`.

Note

`$class = new ReflectionClass('Foo');` `$class->isInstance($arg)` est équivalent à `$arg instanceof Foo` ou `is_a($arg, 'Foo')`.

5.10.18.6 ReflectionMethod

La classe `ReflectionMethod` vous permet de faire du reverse-engineering sur les méthodes des classes.

La classe ReflectionMethod

```

<?php
class ReflectionMethod extends ReflectionFunction
{
    public __construct(mixed class, string name)
    public string __toString()
    public static string export()
    public mixed invoke(stdclass object, mixed* args)
    public mixed invokeArgs(stdclass object, array args)
    public bool isFinal()
    public bool isAbstract()
    public bool isPublic()
    public bool isPrivate()
    public bool isProtected()
    public bool isStatic()
}

```

```

public bool isConstructor()
public bool isDestructor()
public int getModifiers()
public ReflectionClass getDeclaringClass()

// Inherited from ReflectionFunction
final private __clone()
public string getName()
public bool isInternal()
public bool isUserDefined()
public string getFileName()
public int getStartLine()
public int getEndLine()
public string getDocComment()
public array getStaticVariables()
public bool returnsReference()
public ReflectionParameter[] getParameters()
public int getNumberOfParameters()
public int getNumberOfRequiredParameters()
}
?>

```

Pour connaître le fonctionnement d'une méthode, vous devez d'abord créer une instance de la classe `ReflectionMethod`. Vous pourrez ainsi appeler n'importe quelle méthode de cette instance.

Utilisation de la classe `ReflectionMethod`

```

<?php
class Compteur
{
    private static $c = 0;

    /**
     * Incrémentation d'un compteur
     *
     * @final
     * @static
     * @access public
     * @return int
     */
    final public static function increment()
    {
        return self::$c;
    }
}

// Création d'une instance de la classe Reflection_Method
$method = new ReflectionMethod('Compteur', 'increment');

// Affichage d'informations basiques
printf(
    "===> La méthode %s%s%s%s%s%s '%s' (qui est %s)\n".
    "    déclaré dans %s\n".
    "    lignes %d à %d\n".
    "    a les modificateurs %d[%s]\n",
    $method->isInternal() ? 'internal' : 'user-defined',
    $method->isAbstract() ? ' abstract' : '',
    $method->isFinal() ? ' final' : '',
    $method->isPublic() ? ' public' : '',
    $method->isPrivate() ? ' private' : '',
    $method->isProtected() ? ' protected' : '',
    $method->isStatic() ? ' static' : '',
    $method->getName(),
    $method->isConstructor() ? 'the constructor' : 'a regular method',
    $method->getFileName(),
    $method->getStartLine(),

```

```

    $method->getEndline(),
    $method->getModifiers(),
    implode(' ', Reflection::getModifierNames($method->getModifiers()))
);

// Affichage du commentaire de la documentation
printf("----> Documentation:\n %s\n", var_export($method->getDocComment(), 1));

// Affichage des variables statiques si elles existent
if ($statics= $method->getStaticVariables()) {
    printf("----> Variables statiques : %s\n", var_export($statics, 1));
}

// Invocation de la méthode
printf("----> Résultat de l'invocation dans : ");
var_dump($method->invoke(NULL));
?>

```

Note

Invoquer des méthodes privées, protégées ou abstraites provoquera une exception jetée par la méthode `invoke`.

Note

Pour les méthodes statiques comme vu précédemment, vous devez passer NULL comme premier argument à la fonction `invoke`. Pour les méthodes non-statiques, passez une instance de la classe.

5.10.18.7 ReflectionProperty

La classe `ReflectionProperty` vous permet de faire du reverse-engineering sur les propriétés des classes.

La classe ReflectionProperty

```

<?php
class ReflectionProperty implements Reflector
{
    final private __clone()
    public __construct(mixed class, string name)
    public string __toString()
    public static string export()
    public string getName()
    public bool isPublic()
    public bool isPrivate()
    public bool isProtected()
    public bool isStatic()
    public bool isDefault()
    public int getModifiers()
    public mixed getValue(stdclass object)
    public void setValue(stdclass object, mixed value)
    public ReflectionClass getDeclaringClass()
    public string getDocComment()
}
?>

```

Note

`getDocComment` a été ajouté en PHP 5.1.0.

Pour connaître le fonctionnement d'une propriété, vous devez d'abord créer une instance de la classe `ReflectionProperty`. Vous pourrez ainsi appeler n'importe quelles méthodes de cette instance.

Utilisation de la classe ReflectionProperty

```

<?php
class Chaine
{
    public $length = 5;
}

// Création d'une instance de la classe ReflectionProperty
$prop = new ReflectionProperty('Chaine', 'length');

// Affichage d'informations basiques
printf(
    "===> Les propriétés %s%s%s%s '%s' (qui a %s)\n".
    "    a les modificateurs %s\n",
    $prop->isPublic() ? ' public' : '',
    $prop->isPrivate() ? ' private' : '',
    $prop->isProtected() ? ' protected' : '',
    $prop->isStatic() ? ' static' : '',
    $prop->getName(),
    $prop->isDefault() ? 'déclaré au moment de la compilation' : 'créé au moment de l\'exécution',
    var_export(Reflection::getModifierNames($prop->getModifiers()), 1)
);

// Création d'une instance de Chaine
$objj= new Chaine();

// Récupération de la valeur courante
printf("---> La valeur est : ");
var_dump($prop->getValue($objj));

// Modification de la valeur
$prop->setValue($objj, 10);
printf("---> Définition de la valeur à 10, la nouvelle valeur est : ");
var_dump($prop->getValue($objj));

// Affichage de l'objet
var_dump($objj);
?>

```

Note

Essayer de récupérer ou de définir les valeurs des propriétés d'une classe privée ou protégée produira une exception.

5.10.18.8 ReflectionExtension

La classe ReflectionExtension vous permet de faire du reverse-engineering sur les extensions. Vous pouvez connaître toutes les extensions chargées à l'exécution en utilisant la fonction `get_loaded_extensions`.

La classe ReflectionExtension

```

<?php
class ReflectionExtension implements Reflector
{
    final private __clone()
    public __construct(string name)
    public string __toString()
    public static string export()
    public string getName()
    public string getVersion()
    public ReflectionFunction[] getFunctions()
    public array getConstants()
    public array getINIEntries()
    public ReflectionClass[] getClasses()

```

```

    public array getClassNames()
}
?>

```

Pour connaître le fonctionnement d'une extension, vous devez d'abord créer une instance de la classe `ReflectionExtension`. Vous pourrez ainsi appeler n'importe quelle méthode sur cette instance.

Utilisation de la classe `ReflectionExtension`

```

<?php
// Création d'une instance de la classe ReflectionProperty
$ext = new ReflectionExtension('standard');

// Affichage d'informations basiques
printf(
    "Nom          : %s\n".
    "Version       : %s\n".
    "Fonctions      : [%d] %s\n".
    "Entrées INI    : [%d] %s\n" .
    "Classes       : [%d] %s\n",
    $ext->getName(),
    $ext->getVersion() ? $ext->getVersion() : 'NO_VERSION',
    sizeof($ext->getFunctions()),
    var_export($ext->getFunctions(), 1),

    sizeof($ext->getConstants()),
    var_export($ext->getConstants(), 1),

    sizeof($ext->getINIEntries()),
    var_export($ext->getINIEntries(), 1),

    sizeof($ext->getClassNames()),
    var_export($ext->getClassNames(), 1)
);
?>

```

5.10.18.9 Extension des classes de réflexion

Dans le cas où vous voudriez créer des versions spéciales des classes embarquées (par exemple pour créer du HTML colorisé lorsqu'il est exporté, pour avoir un accès facile aux variables des membres au lieu des méthodes ou pour avoir des méthodes utiles), vous devez étendre la classe.

Extension des classes embarquées

```

<?php
/**
 * Ma classe Reflection_Method
 *
 */
class My_Reflection_Method extends ReflectionMethod
{
    public $visibility= '';

    public function __construct($o, $m)
    {
        parent::__construct($o, $m);
        $this->visibility= Reflection::getModifierNames($this->getModifiers());
    }
}

/**
 * Démo classe #1
 *
 */

```

```

*/
class T {
    protected function x() {}
}

/**
 * Démo classe #2
 *
 */
class U extends T {
    function x() {}
}

// Affichage des informations
var_dump(new My_Reflection_Method('U', 'x'));
?>

```

Note

Attention : si vous écrasez le constructeur, n'oubliez pas d'appeler le constructeur parent **avant** d'insérer le moindre code. Sinon, votre code produira l'erreur suivante : Fatal error: Internal error: Failed to retrieve the reflection object

5.10.19 Typage objet

PHP 5 introduit le typage objet implicite (littéralement, Type Hinting). Les fonctions peuvent maintenant imposer aux paramètres d'être des objets (en spécifiant le nom de la classe dans le prototype de la fonction) ou des tableaux (depuis PHP 5.1).

Exemples de typage d'objets

```

<?php
// Un exemple de classe
class MaClasse
{
    /**
     * Fonction de test
     *
     * Le premier paramètre doit être un objet de type AutreClasse
     */
    public function test(AutreClasse $autreclasse) {
        echo $autreclasse->var;
    }

    /**
     * Une autre fonction de test
     *
     * Le premier paramètre doit être un tableau
     */
    public function test_array(array $input_array) {
        print_r($input_array);
    }
}

// Un autre exemple de classe
class AutreClasse {
    public $var = 'Bonjour le monde!';
}
?>

```

Si le paramètre ne satisfait pas les conditions imposées, une erreur fatale est émise.

```

<?php
// Une instance de chaque classe
$maclasse = new MaClasse;
$autreclasse = new AutreClasse;

// Erreur fatale : Argument 1 doit être un objet de la classe AutreClasse
$maclasse->test('salut');

// Erreur fatale : Argument 1 doit être une instance de AutreClasse
$foo = new stdClass;
$maclasse->test($foo);

// Erreur fatale : Argument 1 ne doit pas être null
$maclasse->test(null);

// Fonctionne : Affiche 'Bonjour le monde!'
$maclasse->test($autreclasse);

// Erreur fatale : Argument 1 doit être un tableau
$myclass->test_array('a string');

// Fonctionne : Affiche le tableau
$myclass->test_array(array('a', 'b', 'c'));
?>

```

Le typage fonctionne aussi avec les fonctions :

```

<?php
// Un exemple de classe
class MaClasse {
    public $var = 'Bonjour le monde!';
}

/**
 * Fonction de test
 *
 * Le premier paramètre doit être un objet de type MaClasse
 */
function MaFonction(MaClasse $foo) {
    echo $foo->var;
}

// Fonctionne
$maclasse = new MaClasse;
MaFonction($maclasse);
?>

```

Le typage de paramètre ne fonctionne qu'avec les variables de type object et object . Le typage avec les types traditionnels, tels que int et string , n'est pas supporté.

5.11 Exceptions

PHP 5 a une gestion des exceptions similaire à ce qu'offrent les autres langages de programmation. Une exception peut être jetée, essayée et attrapée dans PHP. Un bloc d'essai doit inclure au moins un bloc de capture. De multiples blocs de capture peuvent être utilisés pour attraper différents types de classes ; l'exception continuera après que le dernier bloc de capture ne soit défini dans la séquence. Les exceptions peuvent être jetées dans les blocs de capture.

Lorsqu'une exception est jetée, le code suivant le traitement ne sera pas exécuté et PHP tentera de trouver le premier bloc de capture correspondant. Si une exception n'est pas attrapée, une erreur

fatale issue de PHP sera envoyée avec un message spécifiant que l'exception n'a pu être attrapée à moins qu'un gestionnaire ne soit défini avec la fonction set_exception_handler .

Lancer une exception

```
<?php
try {
    $error = 'Toujours lancer cette erreur';
    throw new Exception($error);

    // le code suivant une exception n'est pas exécuté.
    echo 'Jamais exécuté';
}
catch (Exception $e) {
    echo "Capture de l'exception : ", $e->getMessage(), "\n";
}

// Continue l'exécution
echo 'Bonjour le Monde !';
?>
```

5.11.1 Exceptions étendues

Une classe Exception définie par l'utilisateur peut être définie en étendant la classe Exception interne. Les membres et les propriétés suivantes montrent ce qui est accessible dans la classe enfant qui est dérivée de la classe exception interne.

La classe Exception interne

```
<?php
class Exception
{
    protected $message = 'exception inconnu'; // message de l'exception
    protected $code = 0; // code de l'exception défini par l'utilisateur
    protected $file; // nom du fichier source de l'exception
    protected $line; // ligne de la source de l'exception

    function __construct(string $message=NULL, int code=0);

    final function getMessage(); // message de l'exception
    final function getCode(); // code de l'exception
    final function getFile(); // nom du fichier source
    final function getLine(); // ligne du fichier source
    final function getTrace(); // un tableau de backtrace()
    final function getTraceAsString(); // chaîne formatée de trace

    /* Remplacable */
    function _toString();
    // chaîne formatée pour l'affichage
}
?>
```

Si une classe étend la classe Exception interne et redéfinit le constructeur , il est vivement recommandé qu'elle appelle aussi parent::__construct() pour s'assurer que toutes les données disponibles ont été proprement assignées. La méthode __toString() peut être réécrite pour fournir un affichage personnalisé lorsque l'objet est présenté comme une chaîne.

Etendre la classe Exception

```
<?php
/**
 * Définition d'une classe d'exception personnalisée
 */
```

```

class MyException extends Exception
{
    // Redéfinissez l'exception ainsi le message n'est pas facultatif
    public function __construct($message, $code = 0) {

        // traitement personnalisé que vous voulez réaliser ...

        // assurez-vous que tout a été assigné proprement
        parent::__construct($message, $code);
    }

    // chaîne personnalisé représentant l'objet
    public function __toString() {
        return __CLASS__ . ": [{".$this->code}]: {".$this->message}\n";
    }

    public function customFunction() {
        echo "Une fonction personnalisée pour ce type d'exception\n";
    }
}

/**
 * Création d'une classe pour tester l'exception
 */
class TestException
{
    public $var;

    const THROW_NONE      = 0;
    const THROW_CUSTOM    = 1;
    const THROW_DEFAULT   = 2;

    function __construct($avalue = self::THROW_NONE) {

        switch ($avalue) {
            case self::THROW_CUSTOM:
                // jète une exception personnalisé
                throw new MyException('1 est un paramètre invalide', 5);
                break;

            case self::THROW_DEFAULT:
                // jète l'exception par défaut.
                throw new Exception('2 n'est pas autorisé en tant que paramètre', 6);

                break;

            default:
                // Aucune exception, l'objet sera créé.
                $this->var = $avalue;
                break;
        }
    }
}

// Exemple 1
try {
    $o = new TestException(TestException::THROW_CUSTOM);
} catch (MyException $e) { // Devrait être attrapée
    echo "Capture mon exception\n", $e;
    $e->customFunction();
} catch (Exception $e) { // Sauté
    echo "Capture l'exception par défaut\n", $e;
}

// Continue l'exécution
var_dump($o);
echo "\n\n";

```

```

//Exemple 2
try {
    $o = new TestException(TestException::THROW_DEFAULT);
} catch (MyException $e) { // Ne correspond pas à ce type
    echo "Capture mon exception\n", $e;
    $e->customFunction();
} catch (Exception $e) { // Devrait être attrapée
    echo "Capture l'exception par défaut\n", $e;
}

// Continue l'exécution
var_dump($o);
echo "\n\n";

// Exemple 3
try {
    $o = new TestException(TestException::THROW_CUSTOM);
} catch (Exception $e) { // Devrait être attrapée
    echo "Capture l'exception par défaut\n", $e;
}

// Continue l'exécution
var_dump($o);
echo "\n\n";

// Exemple 4
try {
    $o = new TestException();
} catch (Exception $e) { // sauté, aucune exception
    echo "Capture l'exception par défaut\n", $e;
}

// Continue l'exécution
var_dump($o);
echo "\n\n";
?>

```

5.12 Les références

5.12.1 Qu'est ce qu'une référence?

En PHP, les références sont destinées à appeler le contenu d'une variable avec un autre nom. Ce n'est pas comme en C ; à la place, les références sont des alias dans la table des symboles. Le nom de la variable et son contenu ont des noms différents, ce qui fait que l'on peut donner plusieurs noms au même contenu. On peut faire l'analogie avec les fichiers sous Unix, et leur nom de fichier : les noms des variables sont les entrées dans un répertoire, tandis que le contenu de la variable est le contenu même du fichier. Faire des références en PHP revient alors à faire des liens sous Unix.

5.12.2 Que font les références ?

Les références vous permettent de faire pointer deux variables sur le même contenu. Par exemple, lorsque vous faites :

Les références

```
<?php
$a =& $b;
?>
```

cela signifie que \$a et \$b pointent sur le même contenu.

Note

\$a et \$b sont complètement égales ici : ce n'est pas \$a qui pointe sur \$b , ou vice versa. C'est bien \$a et \$b qui pointent sur le même contenu.

Note

Si un tableau par référence est copié, ses valeurs ne sont pas déréférencées. Cela est valide également pour les tableaux passés par valeur aux fonctions.

La même syntaxe peut être utilisée avec les fonctions qui retournent des références, et avec l'opérateur new (PHP 4.0.4 et plus récent):

Les références et new()

```
<?php
$bar =& new fooclass();
$foo =& find_var($bar);
?>
```

Depuis PHP 5, new retourne une référence automatiquement, donc, l'utilisation de =& dans ce contexte est déprécié et produit un message de degré E STRICT.

Note

A moins d'utiliser la syntaxe ci-dessus, le résultat de \$bar = new fooclass() ne sera pas la même variable que \$this dans le constructeur, ce qui signifie que si vous avez utilisé la référence \$this dans le constructeur, vous devez assigner la référence, ou bien obtenir deux objets différents.

Même si vous pouvez utiliser l'opérateur @ pour **supprimer** les messages d'erreurs du constructeur avec la syntaxe @new , cela ne fonctionnera pas avec la syntaxe &new . C'est une limitation du moteur Zend, et cela conduit à une erreur d'analyse.

Attention

Si vous assignez une référence à une variable définie en tant que global dans une fonction, la référence sera visible uniquement à l'intérieure de la fonction. Vous pouvez éviter cela en utilisant le tableau \$GLOBALS .

Référencer une variable globale à l'intérieure d'une fonction

```
<?php
$var1 = "Variable Exemple";
$var2 = "";

function global_references($use_globals)
{
    global $var1, $var2;
    if (!$use_globals) {
        $var2 =& $var1; // visible uniquement dans la fonction
    } else {
        $GLOBALS["var2"] =& $var1; // visible également dans un contexte global
    }
}

global_references(false);
echo "var2 est défini à '$var2'\n"; // var2 est défini à ''
global_references(true);
echo "var2 est défini à '$var2'\n"; // var2 est défini à 'Variable Exemple'
?>
```

Voyez global \$var; comme un raccourci pour \$var =& \$GLOBALS['var']; . De ce fait assignant d'autres références à \$var changeant uniquement la référence local de la variable.

Note

Si vous assignez des valeurs par références dans une structure `foreach` , les références seront également modifiées.

Références et structure foreach

```
<?php
$ref = 0;
$row =& $ref;
foreach (array(1, 2, 3) as $row) {
    // faite quelque chose
}
echo $ref; // 3 - le dernier élément du tableau itéré
?>
```

Attention

Les tableaux complexes sont parfois plutôt copiés que référencés. De ce fait, l'exemple suivant ne fonctionnera pas comme prévu :

Références avec un tableau complexe

```
<?php
$top = array(
    'A' => array(),
    'B' => array(
        'B_b' => array(),
    ),
);

$top['A']['parent'] = &$top;
$top['B']['parent'] = &$top;
$top['B']['B_b']['data'] = 'test';
print_r($top['A']['parent']['B']['B_b']); // array()
?>
```

Le deuxième intérêt des références est de pouvoir passer des variables par référence. On réalise ceci en faisant pointer des variables locales vers le contenu des variables de fonction. Exemple :

Passage de paramètre par références

```
<?php
function foo(&$var) {
    $var++;
}
$a=5;
foo($a);
?>
```

\$a vaut 6. Cela provient du fait que dans la fonction foo , la variable \$var pointe sur le même contenu que \$a . Voir aussi les explications détaillées dans [passage par référence](#) .

Le troisième intérêt des références est de [retourner des valeurs par référence](#) .

5.12.3 Ce que les références ne sont pas

Comme précisé ci-dessus, les références ne sont pas des pointeurs. Cela signifie que le script suivant ne fera pas ce à quoi on peut s'attendre :

Les références ne sont pas des pointeurs

```
<?php
function foo(&$var) {
    $var =& $GLOBALS["baz"];
}
foo($bar);
?>
```

Il va se passer que \$var dans foo() sera lié à \$bar , mais il sera aussi relié à \$GLOBALS["baz"] . Il n'y a pas moyen de lier \$bar à quelque chose d'autre en utilisant le mécanisme de référence, car \$bar n'est pas accessible dans la fonction foo() (certes, il est représenté par \$var et \$var possède la même valeur, mais n'est pas relié par la table des symboles).

5.12.4 Passage par référence

Vous pouvez passer des variables par référence, de manière à ce que la fonction modifie ses arguments. La syntaxe est la suivante :

Passage par référence

```
<?php
function foo(&$var) {
    $var++;
}
$a=5;
foo ($a);
// $a vaut 6 maintenant
?>
```

Notez qu'il n'y a pas de signe de référence dans l'appel de la fonction, uniquement sur sa définition. La définition de la fonction est suffisante pour passer correctement des arguments par référence. Dans les versions récentes de PHP, vous devriez recevoir une alerte disant que "Call-time pass-by-reference" est déprécié lorsque vous utilisez un & dans foo(&\$a); .

Les objets suivants peuvent être passés par référence :

- Une variable, i.e. foo(\$a)
- Un nouvel objet, i.e. foo(new foobar())
- Une référence, retournée par une fonction :

Retour d'une référence par une fonction

```
<?php
function &bar() {
    $a = 5;
    return $a;
}
foo(bar());
?>
```

Voir aussi des détails dans [retourner des références](#) .

Toutes les autres expressions ne doivent pas être passées par référence, car le résultat sera indéfini. Par exemple, les passages par référence suivants sont invalides :

Passage par référence invalides

```
<?php
function bar() // Notez l'absence de &
{
    $a = 5;
```

```

    return $a;
}
foo(bar()); // Produit une erreur fatale depuis PHP 5.0.5

foo($a = 5); // Expression, pas une variable
foo(5); // Produit une erreur fatale
?>

```

Ces fonctionnalités sont valables à partir de PHP 4.0.4.

5.12.5 Retourner des références

Retourner des références est toujours utile lorsque vous voulez utiliser une fonction pour savoir à quoi est liée une variable. N'utilisez **pas** le retour par référence pour améliorer les performances, le moteur est suffisamment robuste pour optimiser cela en interne. Retourner uniquement des références lorsque vous avez techniquement une bonne raison de le faire ! Pour retourner des références, utilisez cette syntaxe :

Retourner des références

```

<?php
function &find_var($param) {
    /* ...code... */
    return $found_var;
}
$foo =& find_var ($bar);
$foo->x = 2;
?>

```

Dans cet exemple, on affecte une valeur à la propriété de l'objet retourné par la fonction `find_var`, et non à sa copie, comme ce serait le cas si on n'avait pas utilisé la syntaxe de référence.

Note

Contrairement au passage de paramètre, vous devez utiliser `&` aux deux endroits, à la fois pour indiquer que vous retournez par référence (pas une copie habituelle), et pour indiquer que vous assignez aussi par référence (pas la copie habituelle).

Note

Si vous tentez de retourner une référence depuis une fonction avec la syntaxe : `return ($found_var);`, cela ne fonctionnera **pas** comme vous l'attendez et retournera le résultat de l'**expression**, mais pas de la variable, par référence. Vous ne pouvez retourner des variables par référence que depuis une fonction - rien d'autre. L'erreur `E_NOTICE` est envoyée depuis PHP 4.4.0 et PHP 5.1.0 si le code tente de retourner une expression dynamique ou un résultat de l'opérateur `new`.

5.12.6 Détruire une référence

Lorsque vous détruisez une référence, vous ne faites que casser le lien entre le nom de la variable et son contenu. Cela ne signifie pas que le contenu est détruit. Par exemple :

Détruire une référence

```

<?php
$a = 1;
$b =& $a;
unset($a);
?>

```

Cet exemple ne détruira pas `$b`, mais juste `$a`.

Encore une fois, on peut comparer cette action avec la fonction unlink d'Unix.

5.12.7 Repérer une référence

De nombreuses syntaxes de PHP sont implémentées via le mécanisme de référence, et tout ce qui a été vu concernant les liaisons entre variables s'applique à ces syntaxes. Par exemple, le passage et le retour d'arguments par référence. Quelques autres exemples de syntaxes :

5.12.7.1 Références globales

Lorsque vous déclarez une variable global \$var , vous créez en fait une référence sur une variable globale. Ce qui signifie que

Références sur les variables globales

```
<?php
$var =& $GLOBALS["var"];
?>
```

Et que, si vous détruisez la variable \$var , la variable globale ne sera pas détruite.

5.12.7.2 \$this

Dans une méthode d'objet, \$this est toujours une référence sur l'objet courant.

6 Sécurité

6.1 Introduction

PHP est un langage puissant et l'interpréteur, qu'il soit inclus dans le serveur web ou bien compilé en version CGI, est capable d'accéder aux fichiers, d'exécuter des commandes et d'ouvrir des connexions réseaux. Toutes ces propriétés rendent fragile la sécurité d'un serveur web. Le langage PHP a été pensé afin d'être un langage beaucoup plus sécurisé pour écrire des CGI que le Perl ou le langage C. De plus, une sélection rigoureuse des options de compilation et d'exécution vous permettra d'obtenir un équilibre parfait entre liberté et sécurité.

Etant donné qu'il y a de nombreux moyens d'utiliser le langage PHP, il y a de nombreuses directives de configuration afin d'en contrôler le comportement. Un grand nombre d'options permettent d'utiliser PHP dans de nombreuses situations, mais cela signifie aussi qu'il y a certaines combinaisons d'options de compilation et d'exécution qui fragilisent la sécurité du serveur. Ce chapitre explique comment les différentes options de configuration peuvent être combinées, tout en conservant une sécurité maximum.

La flexibilité de configuration de PHP est épaulée par la flexibilité du code. PHP peut être compilé pour constituer une application serveur complète, avec toutes les fonctionnalités d'un shell, ou il peut encore être utilisé comme simple SSI (server side include) avec peu de risque, dans un environnement à sécurité renforcée. La création de cet environnement et sa sécurité est largement à la charge du développeur PHP.

Ce chapitre commence par expliquer les différentes options de configuration et les situations dans lesquelles elles peuvent être utilisées en toute sécurité. Puis, viennent les considérations de niveaux de sécurité, et les conseils généraux.

6.2 Considérations générales

Un système complètement sûr est une impossibilité virtuelle. L'approche souvent utilisée par les professionnels de la sécurité est d'équilibrer les risques et l'ergonomie. Si chaque variable fournie par l'utilisateur demandait deux formes de validation biométrique (un scan de la rétine et une empreinte digitale), on obtiendrait un système avec un niveau de sécurité d'un bon niveau. Il faudrait aussi une bonne heure pour remplir un formulaire simple, ce qui encouragerait les utilisateurs à trouver un moyen de contourner cette sécurité.

La meilleure sécurité est suffisamment discrète pour assurer un maximum de sécurité sans ajouter de contraintes insurmontables pour l'utilisateur ou de systèmes complexes de programmation. Souvent, les attaques sur un script sont des exploitations des systèmes de sécurité trop complexes, qui s'érodent au cours du temps.

Un principe qu'il est bon de retenir : Un système est aussi sûr que son maillon le plus faible. Si toutes les transactions sont bien notées dans une base, avec confirmation mais que l'utilisateur est identifié uniquement par un cookie, la robustesse de votre système est sévèrement réduite.

Lorsque vous testez votre site, gardez en tête que vous ne pourrez jamais tester toutes les situations, même pour les pages les plus simples. Les valeurs que vous attendez seront toujours complètement différentes des valeurs entrées par un employé négligent, un hacker qui a toute la nuit devant lui ou encore le chat de la maison qui marche sur le clavier. C'est pourquoi il est préférable de regarder le code d'un point de vue logique, pour repérer les points d'entrée des

données inattendues, puis de voir comment elles pourront être modifiées, amplifiées ou réduites.

L'Internet est rempli d'individus qui tentent de se faire une renommée en piratant vos programmes, en bloquant votre site, en envoyant des contenus inappropriés, qui rendent vos journées si "spéciales". Peu importe que vous ayez un grand portail ou un petit web, vous pouvez être la cible pour tout quidam avec une connexion. Vous êtes une cible potentielle dès que vous êtes connecté vous-même. Certains programmes de piratage ne font pas dans la demi-mesure, et testent systématiquement des millions d'IP, à la recherche de victimes : ne soyez pas la prochaine.

6.3 Binaires CGI

6.3.1 Faiblesses connues

Utiliser PHP comme un CGI exécutable vient la majorité du temps du fait que l'on ne veut pas l'utiliser comme un module du serveur web, (comme Apache), ou bien que l'on souhaite l'utiliser en combinaison d'un CGI complémentaire, afin de créer un environnement de script sécurisé (en utilisant des techniques de chroot ou setuid). Une telle décision signifie habituellement que vous installez votre exécutable dans le répertoire cgi-bin de votre serveur web. [CERT CA-96.11](#) recommande effectivement de ne placer aucun interpréteur à l'intérieur du répertoire cgi-bin. Même si le programme PHP peut être utilisé comme interpréteur indépendant, PHP a été pensé afin de rendre impossible les attaques que ce type d'installation induit.

- Accès au système de fichier : `http://ma.machine/cgi-bin/php?/etc/passwd`

Lorsque la requête est passée dans une url, après le point d'interrogation (?), elle est envoyée à l'interpréteur comme une ligne de commande par l'interface CGI. Habituellement, l'interpréteur ouvre le fichier spécifié et l'exécute.

Lorsqu'il est invoqué comme exécutable CGI, PHP refuse d'interpréter les arguments de la ligne de commande.

- Accès d'un document web sur le serveur : `http://my.host/cgi-bin/php/secret/doc.html`

Le "path information" dans l'url, situé juste après le nom de l'exécutable PHP, `/secret/doc.html` est utilisé par convention pour spécifier le nom du fichier qui doit être ouvert et interprété par le programme CGI. Habituellement, des directives de configuration du serveur web (pour le serveur Apache : Action) sont utilisées pour rediriger les requêtes afin d'obtenir un document `http://my.host/secret/script.php` par l'interpréteur PHP. Dans une telle configuration, le serveur web vérifie d'abord s'il a accès au répertoire `/secret`, et après cette vérification redirige la requête vers `http://my.host/cgi-bin/php/secret/script.php`. Malheureusement, si la requête est faite directement sous cette forme, aucune vérification d'accès n'est faite par le serveur web pour le fichier `/secret/script.php`, mais uniquement pour le fichier `/cgi-bin/php`. De cette manière, n'importe quel utilisateur qui peut accéder au fichier `/cgi-bin/php` peut aussi accéder aux documents protégés sur le serveur web.

Avec PHP, l'option de compilation `--enable-force-cgi-redirect` et les options d'exécution `doc_root` et `user_dir` peuvent être utilisées pour prévenir ce genre d'attaques, si des restrictions d'accès sont appliquées sur les documents du serveur. Voir ci-dessous pour des explications plus complètes sur les différentes combinaisons.

6.3.2 Cas 1 : Tous les fichiers sont publics

Si votre serveur n'a aucun document dont l'accès est restreint par un mot de passe ou un système de vérification de l'adresse IP, vous n'avez aucun besoin de ce type de configuration. Si votre serveur web ne permet pas les redirections, ou si votre serveur web n'a aucun besoin de communiquer avec le binaire PHP de manière sécurisée, vous pouvez utiliser l'option de compilation `--disable-force-cgi-redirect`. Vous devez quand même vérifier qu'aucun script ne fait appel au PHP, de manière directe, `http://my.host/cgi-bin/php/dir/script.php` ou bien de manière indirecte, par redirection, `http://my.host/dir/script.php`.

Les redirections peuvent être configurées dans les fichiers de configuration d'Apache en utilisant les directives "AddHandler" et "Action" (voir ci-dessous).

6.3.3 Cas 2 : Utilisation de la directive de compilation `--enable-force-cgi-redirect`

Cette option de compilation prévient quiconque d'appeler directement un script avec l'url `http://my.host/cgi-bin/php/secret/dir/script.php`. Dans ce cas-là, PHP parsera le fichier uniquement s'il y a eu redirection.

Habituellement, le serveur web Apache réalise une redirection grâce aux directives suivantes :

```
Action php-script /cgi-bin/php
AddHandler php-script .php
```

Cette option a uniquement été testée avec Apache et compte sur Apache pour affecter la variable d'environnement non-standard **REDIRECT_STATUS** pour les requêtes redirigées. Dans le cas où votre serveur web ne supporte pas le renseignement de PHP, pour savoir si la requête a été redirigée ou non, vous ne pouvez pas utiliser cette option de compilation. Vous devez alors utiliser une des autres méthodes d'exploitation de la version binaire CGI de PHP, comme exposé ci-dessous.

6.3.4 Cas 3 : Utilisation du "doc_root" ou du "user_dir"

Ajouter un contenu interactif dans votre serveur web, comme des scripts ou des exécutables, est souvent considéré comme une pratique non-sécurisée. Si, par erreur, le script n'est pas exécuté mais affiché comme une page HTML classique, il peut en résulter un vol de propriété intellectuelle ou des problèmes de sécurité à propos des mots de passe notamment. Donc, la majorité des administrateurs préfèrent mettre en place un répertoire spécial pour les scripts qui soit uniquement accessible par le biais du binaire CGI de PHP, et donc, tous les fichiers de ce répertoire seront interprétés et non affichés tels quels.

Aussi, si vous ne pouvez pas utiliser la méthode présentée ci-dessus, il est nécessaire de mettre en place un répertoire "doc_root" différent de votre répertoire "document root" de votre serveur web.

Vous pouvez utiliser la directive `doc_root` dans le fichier de configuration, ou vous pouvez affecter la variable d'environnement **PHP_DOCUMENT_ROOT**. Si cette variable d'environnement est affectée, le binaire CGI de PHP construira toujours le nom de fichier à ouvrir avec `doc_root` et le "path information" de la requête, et donc vous serez sûr qu'aucun script n'est exécuté en dehors du répertoire prédéfini (à l'exception du répertoire désigné par la directive `user_dir` Voir ci-dessous).

Une autre option possible ici est la directive `user_dir`. Lorsque la directive n'est pas activée, seuls les fichiers contenus dans le répertoire `doc_root` peuvent être ouverts. Ouvrir un fichier possédant l'url `http://my.host/~user/doc.php` ne correspond pas à l'ouverture d'un fichier sous le répertoire racine de l'utilisateur mais à l'ouverture du fichier `~user/doc.php` sous le répertoire "doc_root" (oui, un répertoire commence par un tilde [~]).

Si la directive "user_dir" est activée à la valeur `public_php` par exemple, une requête du type `http://my.host/~user/doc.php` ouvrira un fichier appelé `doc.php` sous le répertoire appelé `public_php` sous le répertoire racine de l'utilisateur. Si le répertoire racine des utilisateurs est `/home/user`, le fichier exécuté sera `/home/user/public_php/doc.php`.

`user_dir` et `doc_root` sont deux directives totalement indépendantes et donc vous pouvez contrôler l'accès au répertoire "document root" séparément des répertoires "user directory".

6.3.5 Cas 4 : L'exécutable PHP à l'extérieur de l'arborescence du serveur

Une solution extrêmement sécurisée consiste à mettre l'exécutable PHP à l'extérieur de l'arborescence du serveur web. Dans le répertoire `/usr/local/bin`, par exemple. Le problème de cette méthode est que vous aurez à rajouter la ligne suivante :

Ligne d'invocation de PHP
<code>#!/usr/local/bin/php</code>

dans tous les fichiers contenant des tags PHP. Vous devrez aussi rendre le binaire PHP exécutable. Dans ce cas-là, traitez le fichier exactement comme si vous aviez un autre script écrit en Perl ou en sh ou en un autre langage de script qui utilise `#!` comme mécanisme pour lancer l'interpréteur lui-même.

Pour que l'exécutable PHP prenne en compte les variables d'environnement **PATH_INFO** et **PATH_TRANSLATED** correctement avec cette configuration, vous devez utiliser l'option de compilation `--enable-discard-path`.

6.4 Installé en tant que module Apache

Lorsque PHP est compilé en tant que module Apache, ce module hérite des permissions accordées à l'utilisateur faisant tourner Apache (par défaut, l'utilisateur "nobody"). Cela a plusieurs impacts sur la sécurité et les autorisations. Par exemple, si vous utilisez PHP pour accéder à une base de données, à moins que la base n'ait un système de droits d'accès interne, vous devrez rendre la base accessible à l'utilisateur "nobody". Cela signifie qu'un script mal intentionné peut accéder à la base, la modifier sans identification. Il est aussi possible qu'un robot accède à la page d'administration, et détruise toutes les pages. Vous devez aussi protéger vos bases de données avec les autorisations Apache, ou définir votre propre modèle d'accès avec LDAP, `.htaccess`, etc. et inclure ce code dans tous vos scripts : PHP.

Souvent, lorsqu'on a établi les droits de l'utilisateur PHP (ici, l'utilisateur Apache) pour minimiser les risques, on s'aperçoit que PHP ne peut plus écrire de virus dans les fichiers des utilisateurs. Ou encore, modifier une base de données privée. Il est aussi incapable de modifier des fichiers qu'il devrait pouvoir modifier, ou effectuer certaines transactions.

Une erreur fréquente de sécurité est de donner à l'utilisateur Apache les droits de superadministrateur ou d'améliorer les possibilités d'Apache d'une autre façon.

Donner de telles permissions à l'utilisateur Apache est extrêmement dangereux, et peut compromettre tout le système, telle que l'utilisation des sudo ou du chroot . Pour les novices de la sécurité, une telle utilisation est à exclure d'office.

Il existe des solutions plus simples. En utilisant open_basedir vous pouvez contrôler et restreindre l'accès à certains dossiers qui pourront être utilisés par PHP. Vous pouvez aussi créer des aires de restrictions Apache, pour restreindre les activités anonymes liées aux internautes.

6.5 Sécurité des fichiers

PHP est soumis aux règles de sécurité intrinsèques de la plupart des systèmes serveurs : il respecte notamment les droits des fichiers et des dossiers. Une attention particulière doit être portée aux fichiers ou dossiers qui sont accessibles à tout le monde, afin de s'assurer qu'ils ne divulguent pas d'informations critiques.

Puisque PHP a été fait pour permettre aux utilisateurs d'accéder aux fichiers, il est possible de créer un script qui vous permet de lire des fichiers tels que /etc/passwd, de modifier les connexions ethernet, lancer des impressions de documents, etc. Cela implique notamment que vous devez vous assurer que les fichiers accédés par les scripts sont bien ceux qu'il faut.

Considérez le script suivant, où l'utilisateur indique qu'il souhaite effacer un fichier dans son dossier racine. Nous supposons que PHP est utilisé comme interface web pour gérer les fichiers, et que l'utilisateur Apache est autorisé à effacer les fichiers dans le dossier racine des utilisateurs.

Une erreur de vérification de variable conduit à ...

```
<?php
// Efface un fichier dans un dossier racine
$username = $_POST['user_submitted_name'];
$homedir = "/home/$username";
$file_to_delete = "$userfile";
unlink ($homedir/$userfile);
echo "$file_to_delete a été effacé!";
?>
```

Etant donné que le nom de l'utilisateur est à fournir, des intrus peuvent envoyer un nom d'utilisateur autre que le leur, et effacer des documents dans les comptes des autres utilisateurs. Dans ce cas, vous souhaitez utiliser une autre forme d'identification. Considérez ce qui pourrait se passer si les utilisateurs passent `../etc/` et `passwd` comme arguments! Le code serait exécuté tel que :

Une attaque du système de fichiers!

```
<?php
// efface un fichier n'importe où sur le disque dur,
// où l'utilisateur PHP a accès. Si PHP a un accès root :
$username = "../etc/";
$homedir = "/home/../etc/";
$file_to_delete = "passwd";
unlink ("/home/../etc/passwd");
echo "/home/../etc/passwd a été effacé!";
?>
```

Il y a deux mesures primordiales à prendre pour éviter ces manœuvres :

- Limiter les permissions de l'utilisateur web PHP.
- Vérifier toutes les variables liées aux chemins et aux fichiers qui sont fournis.

Voici un script renforcé :

Une vérification renforcée

```
<?php
// Efface un fichier sur le disque où l'utilisateur a le droit d'aller
$username = $_SERVER['REMOTE_USER'];
// utilise un mécanisme d'authentification
$homedir = "/home/$username";
$file_to_delete = basename("$userfile");
// supprime le chemin excédentaire
unlink ($homedir/$file_to_delete);
$fp = fopen("/home/logging/filedelete.log","a"); //note l'effacement
$logstring = "$username $homedir $file_to_delete";
fputs ($fp, $logstring);
fclose($fp);
echo "$file_to_delete a été effacé!";
?>
```

Cependant, même cette technique n'est pas sans faille. Si votre système d'identification permet aux utilisateurs de créer leur propre login, et qu'un utilisateur choisi le login `../etc/`, le système est de nouveau exposé. Pour cette raison, vous pouvez essayer d'écrire un script renforcé :

Vérification renforcée de noms de fichiers

```
<?php
$username = $_SERVER['REMOTE_USER'];
$homedir = "/home/$username";
if (!ereg('^[^./][^/]*$', $username))
    die('Erreur de nom de fichier');
//meurt, ne SURTOUT pas traiter!
// etc.
?>
```

Suivant votre système d'exploitation, vous devrez protéger un grand nombre de fichiers, notamment les entrées de périphériques (`/dev/` ou `COM1`), les fichiers de configuration (fichiers `/etc/` et `.ini`), les lieux de stockage d'informations (`/home/`, `My Documents`), etc. Pour cette raison, il est généralement plus sûr d'établir une politique qui interdit TOUT sauf ce que vous autorisez.

6.6 Sécurité des bases de données

De nos jours, les bases de données sont des composants incontournables des serveurs web et des applications en ligne, qui fournissent du contenu dynamique. Des données secrètes ou critiques peuvent être stockées dans les bases de données : il est donc important de les protéger efficacement.

Pour lire ou stocker des informations, vous devez vous connecter au serveur de bases de données, envoyer une requête valide, lire le résultat et refermer la connexion. De nos jours, le langage le plus courant pour ce type de communication est le langage SQL (Structured Query Language). Voyez comment un pirate peut s'introduire dans une requête SQL .

Comme vous pouvez le réaliser, PHP ne peut pas protéger vos bases de données pour vous. La section suivante vous introduira aux notions de base pour protéger vos bases de données, lors de la programmation de vos scripts.

Gardez bien cette règle simple en tête : la défense se fait par couches. Plus vous ajouterez de tests pour protéger votre base, plus faible sera la probabilité de réussite d'un pirate. Ajoutez à cela un bon schéma de base de données, et vous aurez une application réussie.

6.6.1 Schéma de base de données

La première étape est de créer une base de données, à moins que vous ne souhaitiez utiliser une base de données déjà créée. Lorsque la base de données est créée, un utilisateur propriétaire en est responsable. Généralement, seul le propriétaire (et le super utilisateur) peuvent intervenir avec les tables de cette base, et il faut que ce dernier donne des droits à tous les intervenants qui auront à travailler sur cette base.

Les applications ne doivent jamais se connecter au serveur de bases de données sous le nom du propriétaire ou de l'administrateur, car ces utilisateurs ont des droits très importants, et pourront exécuter n'importe quelle requête, comme la modification de tables, l'effacement de lignes ou même encore, la destruction de la base.

Vous pouvez créer différents utilisateurs de bases de données pour chaque aspect de votre application, avec des droits limités aux seules actions planifiées. Il faut alors éviter que le même utilisateur dispose des droits de plusieurs cas d'utilisation. Cela permet que si des intrus obtiennent l'accès à la base avec l'un de ces jeux de droits, ils ne pourront pas affecter toute l'application.

Il est recommandé de ne pas implémenter toute la logique fonctionnelle dans l'application web (c'est-à-dire dans vos scripts), mais d'en reporter une partie dans la base en utilisant les triggers, vues et règles. Si le système évolue, les nouvelles versions vous feront réécrire toute la logique et donc tous vos scripts. De plus, l'utilisation de trigger permet de gérer de manière transparente des données, et fournit des indications pour déboguer votre application.

6.6.2 Connexions au serveur de base de données

Il est recommandé d'établir des connexions au serveur avec le protocole SSL, pour chiffrer les échanges clients/serveur, afin d'améliorer la sécurité. Vous pouvez aussi utiliser un client SSH pour chiffrer la connexion entre les clients et le serveur de bases de données. Si l'une de ces deux protections est mise en place, il sera difficile de surveiller votre trafic et de comprendre les informations échangées.

6.6.3 Modèle de stockage avec chiffrement

Les protocoles SSL/SSH protègent les données qui circulent entre le serveur et le client, mais SSL/SSH ne protègent pas les données une fois dans la base. SSL est un protocole en ligne.

Une fois que le pirate a obtenu l'accès direct à votre base de données (en contournant le serveur web), les données sensibles, stockées dans votre base sont accessibles directement, à moins que les données de la base ne soient protégées par la base. Chiffrer les données est une bonne solution pour réduire cette menace, mais très peu de bases de données offrent ce type de chiffrement.

Le moyen le plus simple pour contourner ce problème est de créer votre propre logiciel de chiffrement, et de l'utiliser dans vos scripts PHP. PHP peut vous aider dans cette tâche grâce aux nombreuses extensions dont il dispose, comme [Mcrypt](#) et [Mhash](#), qui connaissent un large éventail de méthodes de chiffrement. Le script PHP va chiffrer les données qui seront stockées, et les déchiffrer lorsqu'elles seront relues. Voyez la suite pour des exemples d'utilisation de ce chiffrement.

Dans le cas de données vraiment sensibles, si la représentation originale n'est pas nécessaire (pour affichage, ou comparaison), utiliser un hash est une bonne solution. L'exemple classique est

le stockage de mots de passe dans les bases de données, après les avoir passé au MD5. Voyez les fonctions [crypt](#) et [md5](#) .

Utiliser un mot de passe et MD5

```
<?php
// Stockage du mot de passe hashé
$query = sprintf("INSERT INTO users(name,pwd) VALUES('%s','%s');",
    addslashes($username), md5($password));
$result = pg_query($connection, $query);

// interroger le serveur pour comparer le mot de passe soumis
$query = sprintf("SELECT 1 FROM users WHERE name='%s' AND pwd='%s';",
    addslashes($username), md5($password));
$result = pg_query($connection, $query);

if (pg_num_rows($result) > 0) {
    echo "Bienvenue, $username!";
} else {
    echo "Identification échouée pour $username.";
}
?>
```

6.6.4 Injection SQL

De nombreux développeurs web ne sont pas conscients des possibilités de manipulation des requêtes SQL, et supposent que les requêtes SQL sont des commandes sûres. Cela signifie qu'une requête SQL est capable de contourner les contrôles et vérifications, comme les identifications, et parfois, les requêtes SQL ont accès aux commandes d'administration.

L'injection SQL directe est une technique où un pirate modifie une requête SQL existante pour afficher des données cachées, ou pour écraser des valeurs importantes, ou encore exécuter des commandes dangereuses pour la base. Cela se fait lorsque l'application prend les données envoyées par l'internaute, et l'utilise directement pour contruire une requête SQL. Les exemples ci-dessous sont basés sur une histoire vraie, malheureusement.

Avec le manque de vérification des données de l'internaute et la connexion au serveur avec des droits de super utilisateur, le pirate peut créer des utilisateurs, et créer un autre super utilisateur.

Séparation des résultats en pages, et créer des administrateurs (PostgreSQL et MySQL)

```
<?php
$offset = $argv[0]; // Attention, aucune validation!
$query = "SELECT id, name FROM products ORDER BY name LIMIT 20 OFFSET $offset;";
// avec PostgreSQL
$result = pg_query($conn, $query);
// avec MySQL
$result = mysql_query($query);
?>
```

Un utilisateur normal clique sur les boutons 'suivant' et 'précédent', qui sont alors placés dans la variable \$offset , encodée dans l'URL. Le script s'attend à ce que la variable \$offset soit alors un nombre décimal. Cependant, il est possible de modifier l'URL en ajoutant une nouvelle valeur, au format URL, comme ceci :

Exemple d'injection SQL

```
<?php
// cas de PostgreSQL
0;
insert into pg_shadow(username,usesysid,usesuper,usecatupd,passwd)
select 'crack', usesysid, 't','t','crack'
```

```

    from pg_shadow where username='postgres';
--
// cas de MySQL
0;
UPDATE user SET Password=PASSWORD('crack') WHERE user='root';
FLUSH PRIVILEGES;
?>

```

Si cela arrive, le script va créer un nouveau super utilisateur. Notez que la valeur 0; sert à terminer la requête originale et la terminer correctement.

Note

C'est une technique répandue que de forcer l'analyseur SQL à ignorer le reste de la requête, en utilisant les symboles -- pour mettre en commentaires.

Un moyen disponible pour accéder aux mots de passe est de contourner la recherche de page. Ce que le pirate doit faire, c'est simplement voir si une variable du formulaire est utilisée dans la requête, et si elle est mal gérée. Ces variables peuvent avoir été configurées dans une page précédente pour être utilisées dans les clauses WHERE, ORDER BY, LIMIT et OFFSET des requêtes SELECT . Si votre base de données supporte les commandes UNION , le pirate peut essayer d'ajouter une requête entière pour lister les mots de passe dans n'importe quelle table. Utiliser la technique des mots de passe chiffrés est fortement recommandé.

Liste d'articles ... et ajout de mot de passe

```

<?php
$query = "SELECT id, name, inserted, size FROM products
        WHERE size = '$size'
        ORDER BY $order LIMIT $limit, $offset;";
$result = odbc_exec($conn, $query);
?>

```

La partie statique de la requête, combinée avec une autre requête SELECT , va révéler les mots de passe :

Révélation des mots de passe

```

<?php
'
union select '1', concat(uname||'-'||passwd) as name, '1971-01-01', '0' from usertable;
--
?>

```

Si cette requête, exploitant les ' et -- est affectée à une variable utilisée dans \$query , une injection SQL va se produire.

Les commandes UPDATE sont aussi sujettes à des attaques de votre base de données. Ces requêtes peuvent aussi introduire toute une nouvelle requête dans votre commande initiale. Mais en plus, le pirate peut jouer sur la commande SET . Dans ce cas, il doit connaître un peu votre base de données. Cela peut se deviner en examinant les noms de variables dans les formulaires, ou simplement, en testant les cas les plus classiques. Il n'y a pas beaucoup de conventions de noms pour stocker des noms d'utilisateurs et des mots de passe.

Modifier un mot de passe ... et gain de droits!

```

<?php
$query= "UPDATE usertable SET pwd='$pwd' WHERE uid='$uid';";
?>

```

Mais un internaute fourbe peut envoyer une valeur telle que ' or uid like '%admin%'; -- dans \$uid pour modifier le mot de passe utilisateur, ou simplement, utiliser la variable \$pwd avec la valeur "hehehe", admin='yes', trusted=100 " (avec l'espace final) pour obtenir des droits supplémentaires.

La requête devient alors :

Une requête et son injection

```
<?php
// $uid == ' or uid like '%admin%'; --
$query = "UPDATE usertable SET pwd='...' WHERE uid='' or uid like '%admin%'; --";

// $pwd == "hehehe", admin='yes', trusted=100 "
$query = "UPDATE usertable SET pwd='hehehe', admin='yes', trusted=100 WHERE ...;"
?>
```

C'est un exemple terrible d'acquisition de droits d'administrateur sur un serveur de base de données.

Attaque d'un serveur de bases de données (MSSQL Server)

```
<?php
$query = "SELECT * FROM products WHERE id LIKE '%$prod%'";
$result = mssql_query($query);
?>
```

Si le pirate injecte la valeur `a%' exec master..xp_cmdshell 'net user test testpass /ADD' --` dans la variable `$prod`, alors la requête `$query` devient :

Attaque d'un serveur de base de données (MSSQL Server) - 2

```
<?php
$query = "SELECT * FROM products
          WHERE id LIKE 'a%'
          exec master..xp_cmdshell 'net user test testpass /ADD'--";
$result = mssql_query($query);
?>
```

MSSQL Server exécute les requêtes SQL en lot, y compris la commande d'ajout d'un nouvel utilisateur à la base de données locale. Si cette application fonctionnait en tant que sa et que le service MSSQLSERVER disposait de niveau de droits suffisant, le pirate dispose désormais d'un compte avec accès au serveur.

Note

Certains des exemples ci-dessus sont spécifiques à certains serveurs de bases de données. Cela n'empêche pas des attaques similaires d'être possibles sur d'autres produits. Votre base de données sera alors vulnérable d'une autre manière.

6.6.4.1 Techniques de contournement

Vous pouvez prétendre que le pirate doit d'abord obtenir des informations sur le schéma de la base de données, dans la plupart des cas d'injections. C'est vrai, mais vous ne saurez jamais comment ni quand ces informations auront filtré, et si cela arrive, votre base de données sera en grand danger. Si vous utilisez une base de données Open Srouce, ou une base qui est du domaine public, ou encore un schéma qui appartient à un système de gestion de contenu ou d'un forum, le pirate peut facilement se procurer une copie du code que vous utilisez. Cela peut être un risque potentiel si la base a été mal conçue.

Ces attaques sont généralement basées sur l'exploitation de code qui n'est pas écrit de manière sécuritaire. N'ayez aucune confiance dans les données qui proviennent de l'utilisateur, même si cela provient d'un menu déroulant, d'un champ caché ou d'un cookie. Le premier exemple montre comment une requête peut causer un désastre.

- Ne nous connectez jamais sur une base de données en tant que super utilisateur ou propriétaire de la base. Utilisez toujours un utilisateur adapté, avec des droits très limités.
- Vérifiez que les données ont bien le type attendu. PHP dispose d'un éventail de fonction de

validation large, depuis les plus simples, de la section [Variables](#) et la section [Caractères](#) (e.g. [is_numeric](#) , [ctype_digit](#) respectivement) aux fonctions avancées de [Expression rationnelle Perl](#) .

- Si l'application attend une entrée numérique, vérifiez vos données avec la fonction [is_numeric](#) , ou bien modifiez automatiquement le type avec la fonction [settype](#) , ou encore avec [sprintf](#) .

Une navigation de fiches plus sécuritaire

```
<?php
settype($offset, 'integer');
$query = "SELECT id, name FROM products ORDER BY name LIMIT 20 OFFSET $offset;";

// notez que %d dans la chaîne de format : %s serait inutile
$query = sprintf("SELECT id, name FROM products ORDER BY name LIMIT 20 OFFSET %d",
                $offset);
?>
```

- Mettez entre guillemets toutes les valeurs non numériques qui sont passées à la base de données avec la fonction spécifique à la base de données d'échappement de caractères (e.g. [mysql_escape_string](#) , [sql_escape_string](#) , etc.). Si un mécanisme d'échappement spécifique à une base de données n'existe pas, les fonctions [addslashes](#) et [str_replace](#) peuvent être très utiles (en fonction du type de la base de données). Lisez [le premier exemple](#) . Comme le montre l'exemple, ajouter des guillemets à la partie statique de la requête n'est pas suffisant, rendant cette requête facilement piratable.
- N'affichez jamais d'informations spécifiques à la base, et notamment des informations concernant le schéma. Voyez aussi la section [Rapport d'erreur](#) et le chapitre [Gestion des erreurs](#) .
- Vous pouvez avoir des procédures stockées et des curseurs prédéfinis qui font que les utilisateurs n'ont pas un accès direct aux tables ou vues, mais cette solution a d'autres impacts.

A côté de ces conseils, il est recommandé d'enregistrer vos requêtes soit dans vos scripts soit dans la base elle-même, si elle le supporte. Evidemment, cet enregistrement ne sera pas capable d'empêcher une attaque, mais vous permettra de retrouver la requête qui a fauté. L'historique n'est pas très utile par lui-même, mais au niveau des informations qu'il contient. Plus vous avez de détails, mieux c'est.

6.7 Rapport d'erreurs

En termes de sécurité, il y a deux conséquences au rapport d'erreur. D'un côté, cela améliore la sécurité, mais d'un autre, cela la réduit aussi.

Une tactique d'attaque standard consiste à faire faire des erreurs au système, et lire les variables d'environnement et de contexte qui sont retournées. Cela permet au pirate de lire de nombreuses informations sur le serveur, et de détecter des faiblesses du serveur. Par exemple, si un intrus a glané des informations sur votre page, avec une première utilisation de votre site, il peut essayer de remplacer les variables par ses propres valeurs :

Attaque de site avec une page HTML personnalisée

```
<form method="post" action="http://www.site.cible.com/?username=badfoo&password=badfoo">
<input type="hidden" name="username" value="badfoo">
<input type="hidden" name="password" value="badfoo">
</form>
```

Les erreurs PHP qui sont normalement retournées peuvent être très pratiques pour un développeur qui essaie de déboguer un script, car elles donnent de précieux renseignements tels que quelle

fonction a échoué, quel fichier n'a pas été trouvé, quel script PHP a bogé, et quelle ligne est en faute. Toutes ces informations sont exploitables. Il est commun aux développeurs PHP d'utiliser les fonctions `show_source`, `highlight_string`, ou `highlight_file` comme outils de débogage, mais sur un site en production, cela expose des variables cachées, des syntaxes non vérifiées ou d'autres informations critiques. Il est particulièrement dangereux d'exécuter du code de sources connues, avec les gestionnaires de débogage. Si l'intrus peut comprendre votre technique habituelle d'utilisation, il peut tenter une attaque frontale sur une page, en envoyant des chaînes de débogage :

Exploiter des variables classiques de débogage

```
<form method="post" action="http://www.site.cible.com/?errors=Y&showerrors=1&debug=1">
<input type="hidden" name="errors" value="Y">
<input type="hidden" name="showerrors" value="1">
<input type="hidden" name="debug" value="1">
</form>
```

Indépendamment de la gestion des erreurs, la possibilité de tester la gestion des erreurs d'un système conduit à un trou de sécurité, et la diffusion de plus d'informations sur votre système.

Si un pirate affiche une page HTML, et essaye de la tester (pour rechercher des faiblesses du système), il peut déterminer sur quel système PHP a été compilé.

Une erreur de fonction indique si un système supporte une base de données spécifique, ou bien indique comment la page a été générée. Cela peut orienter l'intrus vers les ports de cette base de données ou bien vers une attaque liée à cette application. En envoyant des données erronées, par exemple, un pirate peut déterminer l'ordre d'identification dans un script, à partir des lignes d'erreurs, et essayer de les exploiter ailleurs, dans le script.

Une erreur de fichier, ou une erreur générale PHP peut indiquer quelles sont les permissions du serveur web, ainsi que la structure et l'organisation des fichiers. Les gestionnaires d'erreurs utilisateurs peuvent aussi aggraver ce problème, en permettant l'exploitation facile d'informations préalablement cachées.

Il y a trois solutions majeures à ces problèmes : la première est de scruter toutes les fonctions, et d'essayer de traiter toutes les erreurs. La deuxième est de désactiver le rapport d'erreur, dès que le script est en production. La troisième est d'utiliser les fonctions de gestion des erreurs. Suivant votre politique de sécurité, vous pouvez utiliser un panachage savant des trois méthodes.

Une méthode pour gagner du temps est d'utiliser la fonction `error_reporting`, pour vous aider à sécuriser le code, et détecter les utilisations dangereuses de variables. Vous testez votre code en bêta-test avec la valeur `E_ALL`, et vous pouvez rapidement repérer les variables qui ne sont pas protégées. Une fois que le code est prêt à être déployé, vous devez soit désactiver complètement le report d'erreur en affectant 0 à la fonction `error_reporting`, soit en désactivant l'affichage des erreurs en utilisant l'option de configuration `display_errors` du `php.ini`. Si vous choisissez la dernière solution, vous devez également définir le chemin de votre fichier de log en utilisant la directive de configuration `error_log`, et en activant la directive `log_errors`.

Détecter des variables non protégées avec `E_ALL`

```
<?php
  if ($username) {
// Non initialisée, ou vérifiée avant utilisation
    $good_login = 1;
  }
  if ($good_login == 1) {
// Si le test ci-dessus échoue, les valeurs n'ont pas été testées
    fpassthru ("/données/très/très/sensibles/index.html");
  }
}
```

?>

6.8 Utilisation des variables super-globales

L'une des évolutions les plus controversées de PHP a été le changement de valeur par défaut de la directive PHP `register_globals`, qui est passée de On à Off en PHP 4.2.0. Beaucoup d'applications dépendaient de cette directive, et de nombreux programmeurs ne savaient même pas qu'elle existait, et supposait que c'était le fonctionnement normal de PHP. Cette page explique comment on peut écrire du code peu sécuritaire en utilisant cette directive. Gardez bien en tête que cette directive, par elle-même, n'est pas un trou de sécurité, mais qu'elle facilite leur création.

Lorsqu'elle est activée, `register_globals` va injecter vos scripts avec toutes sortes de variables, comme les variables issues des formulaires HTML. Ceci, couplé au fait que PHP ne requiert pas d'initialisation de variable signifie que la programmation de script peu sûr est possible. Ce fut une décision difficile de la communauté PHP, mais finalement, il a été décidé de désactiver par défaut cette variable. Lorsqu'elle est active, le programmeur ne sait pas exactement d'où provient le contenu de la variable, et ne peut que faire des suppositions. Les variables internes définies dans le script sont mélangées avec les données envoyées par les utilisateurs, et en désactivant `register_globals`, on empêche cela. Voyons avec un exemple le fonctionnement de `register_globals` :

Exemple de mésutilisation de `register_globals = on`

```
<?php
// $authorized = true uniquement si l'utilisateur est identifié
if (authenticated_user()) {
    $authorized = true;
}

// Comme nous n'avons pas initialisé $authorized avec false, cette dernière
// peut être définie via register_globals, comme avec l'URL GET auth.php?authorized=1
// Tout le monde peut facilement être reconnu comme identifié!
if ($authorized) {
    include "/donnees/critiques/data.php";
}
?>
```

Lorsque `register_globals` est activé, la logique ci-dessus peut être prise en défaut. Lorsque `register_globals` est désactivée `$authorized` ne peut plus être assignée via la requête, et le script est maintenant sûr, même s'il reste recommandé de toujours initialiser ses variables. Par exemple, dans notre programme ci-dessus, nous pourrions ajouter `$authorized = false`. En faisant cela, le script peut fonctionner avec `register_globals on` ou `off`, car les utilisateurs seront par défaut non-identifiés.

Un autre exemple implique les `sessions`. Lorsque `register_globals` est activé, on peut aussi utiliser `$username` dans notre exemple, mais, encore une fois, vous devez garder en tête que `$username` peut aussi provenir d'autres biais, tels que GET (via l'URL).

Exemple d'abus de sessions avec `register_globals on` ou `off`

```
<?php
// Nous ne savons pas d'où provient $username mais nous savons que
// $_SESSION contient les données de session
if (isset($_SESSION['username'])) {
    echo "Bonjour <strong>".$_SESSION['username']</strong>";
} else {
```

```

    echo "Bonjour <strong>visiteur</strong><br />";
    echo "Voulez-vous vous identifier?";
}
?>

```

Il est même possible de prendre des mesures préventives, pour être alerté lorsqu'une tentative d'usurpation est faite. Si vous savez à l'avance de quelle variable le nom d'utilisateur doit provenir, vous pouvez vérifier si les données que vous manipulez sont d'une origine contrôlée. Même si cela ne garantit pas que les données ne puissent être falsifiées, cela complique la tâche du faussaire. Si vous ne vous préoccupez pas de l'origine des données, vous pouvez utiliser la variable `$_REQUEST` qui contient un mélange de données GET, POST et COOKIE. Voyez aussi la section du manuel concernant les [variables externes à PHP](#) .

Détection simple de fausses variables

```

<?php
if (isset($_COOKIE['MAGIC_COOKIE'])) {

    // MAGIC_COOKIE provient d'un cookie.
    // Assurez-vous de valider les données du cookie!

} elseif (isset($_GET['MAGIC_COOKIE']) || isset($_POST['MAGIC_COOKIE'])) {

    mail("admin@example.com", "Tentative possible d'attaque", $_SERVER['REMOTE_ADDR']);
    echo "Alerte sécurité, l'admin a été prévenu.";
    exit;

} else {

    // MAGIC_COOKIE ne provient pas de REQUEST

}
?>

```

Bien sur, désactiver l'option `register_globals` ne signifie pas que votre code est sécuritaire. Pour chaque donnée reçu, vous devez appliquer un maximum de validations. Vérifiez toujours les données de votre visiteur, et initialisez vos variables! Pour vérifier les variables non-initialisées, voyez la fonction [error_reporting](#) , qui peut afficher les erreurs de niveau `E_NOTICE` .

Pour des informations sur l'émulation de `register_globals` On ou Off, voyez la [FAQ](#) .

Note

Superglobales : disponibilité

Depuis PHP 4.1.0, les tableaux superglobaux tels que `$_GET` , `$_POST` et `$_SERVER` , etc. sont disponibles. Pour plus d'informations, lisez la section [superglobaux](#)

6.9 Données transmises par les internautes

Les plus grandes faiblesses de nombreux programmes PHP ne viennent pas du langage lui-même, mais de son utilisation en omettant les caractéristiques de sécurité. Pour cette raison, vous devez toujours prendre le temps de prendre en compte les implications d'une fonction, et de cerner toutes les applications d'une utilisation exotiques des paramètres.

Utilisation dangereuse de variables

```

<?php
// efface un fichier à la racine d'un utilisateur... ou peut être
// de quelqu'un d'autre?

```

```

    unlink($evil_var);
// Note l'accès de ce fichier ... ou pas?
    fputs($fp, $evil_var);
// Exécute une commande triviale... ou ajoute une entrée dans /etc/password ?
    system($evil_var);
    exec($evil_var);
?>

```

Il est vivement recommandé d'examiner minutieusement votre code pour vous assurer qu'il n'y a pas de variables envoyées par le client web, et qui ne sont pas suffisamment vérifiées avant utilisation.

- Est-ce que ce script n'affectera que les fichiers prévus?
- Est-il possible que des valeurs incohérentes soient exploitées ici?
- Est-ce que ce script peut être utilisé dans un but différent?
- Est-ce que ce script peut être utilisé malicieusement, en conjonction avec d'autres?
- Est-ce que toutes les actions seront notées?

En répondant de manière adéquate à ces questions lors de l'écriture de vos scripts (plutôt qu'après), vous éviterez une réécriture inopportune pour raison de sécurité. En commençant vos projets avec ces recommandations en tête, vous ne garantirez pas la sécurité de votre système, mais vous contribuerez à l'améliorer.

Vous pouvez aussi envisager de supprimer l'acquisition automatique des variables d'environnement, les guillemets magiques (magic_quotes), ou encore toute option qui pourrait vous conduire à mésévaluer la validité, la source ou la valeur d'une variable. En travaillant avec `error_reporting(E_ALL)`, vous pouvez être averti que certaines variables sont utilisées avant d'être exploitées, ou vérifiées (et donc, vous pourrez traiter des valeurs exotiques à la source).

6.10 Guillemets magiques

Les guillemets magiques (littéralement, les Magic Quotes) est un processus qui protège automatiquement les données introduites dans un script PHP. Il est recommandé de développer les applications sans l'aide des guillemets magiques, et de protéger manuellement les données.

6.10.1 Qu'est-ce que les guillemets magiques?

Lorsque cette directive est active, les guillemets simples ' , les guillemets doubles " , les anti-slashes \ et les caractères null NULL sont protégés automatiquement avec un anti-slash. C'est le même résultat que celui de la fonction [addslashes](#) .

Il y a trois directives de guillemets magiques différentes :

- [magic_quotes_gpc](#)

Affecte les données issues des requêtes HTTP (GET, POST, et COOKIE). Ne peut pas être configurée durant l'exécution, et vaut par défaut **on** en PHP.

Voir aussi [get_magic_quotes_gpc](#) .

- [magic_quotes_runtime](#)

Si activée, la plupart des fonctions qui retournent des données externes, y compris issues d'une base de données ou d'un fichier texte, verront les données protégées par des

anti-slash. Cette directive peut être modifiée durant l'exécution, et vaut par défaut **off** en PHP

Voir aussi [set_magic_quotes_runtime](#) et [get_magic_quotes_runtime](#) .

- [magic_quotes_sybase](#)

Si cette configuration est active, les guillemets simples sont protégés avec un autre guillemets simples, et non pas un anti-slash. Lorsqu'elle est active, cette directive remplace entièrement [magic_quotes_gpc](#) . Si vous activez ces deux directives, alors seuls les guillemets simples seront protégés, avec " . Les guillemets doubles, les anti-slash et les caractères null seront laissés intacts.

Voir aussi [ini_get](#) pour lire la valeur de la directive.

6.10.2 Pourquoi utiliser les guillemets magiques?

- Utile pour les débutants

Les guillemets magiques sont implantés en PHP pour protéger le code contre les trous de sécurités. Même si les [injections SQL](#) sont possibles avec les guillemets magiques, le risque est réduit.

- Commodité

Lors de l'insertion de données dans une base de données, les guillemets magiques utilisent la fonction [addslashes](#) sur toutes les données de type GET, POST et COOKIE, automatiquement.

6.10.3 Pourquoi ne pas utiliser les guillemets magiques?

- Portabilité

Cette directive peut être activée ou désactivée suivant les serveurs et cela affecte grandement la portabilité. Utilisez [get_magic_quotes_gpc](#) pour vérifier s'ils sont actifs ou pas, et adaptez votre application.

- Performances

Comme ce n'est pas toutes les données qui sont finalement placées dans une base, il y a un coût en vitesse pour protéger toutes ces données. Le simple appel des fonctions de protections en fonction des besoins est plus efficace ([addslashes](#)).

Même si php.ini-dist active ces options par défaut, php.ini-recommended les désactive. Cette recommandation est surtout faite pour des raisons de vitesse.

- Peu pratique

Comme toutes les données n'ont pas forcément besoin de protection, il est souvent désagréable de voir des données protégées là où ça ne sert à rien. Par exemple, lorsque vous envoyez par mail un formulaire, et que vous voyez des anti-slashes parsemer le message. Pour corriger cela, il faut faire un usage fréquent de [stripslashes](#) .

6.10.4 Désactiver les guillemets magiques

La directive `magic_quotes_gpc` ne peut être désactivée qu'au niveau du système, et non pas à l'exécution. En d'autres termes, utiliser `ini_set` n'est pas possible.

Désactiver les guillemets magiques du côté du serveur

Voici un exemple qui donne la valeur de Off à ces directives dans le fichier `php.ini`. Pour plus de détails, voyez la section [Comment changer la configuration](#).

```
; Magic quotes
;

; Magic quotes for incoming GET/POST/Cookie data.
magic_quotes_gpc = Off

; Magic quotes for runtime-generated data, e.g. data from SQL, from exec(), etc.
magic_quotes_runtime = Off

; Use Sybase-style magic quotes (escape ' with " instead of \').
magic_quotes_sybase = Off
```

Si vous n'avez pas accès au serveur, utilisez le fichier `.htaccess`. Par exemple :

```
php_flag magic_quotes_gpc Off
```

Afin d'écrire du code portable sur tous les environnements, et où vous ne pourrez pas forcément modifier la configuration du serveur, voici un exemple de désactivation de `magic_quotes_gpc` à l'exécution. Cette méthode est inefficace, et il est recommandé d'utiliser les autres solutions si possible.

Désactivation des guillemets magiques à l'exécution

```
<?php
if (get_magic_quotes_gpc()) {
    function stripslashes_deep($value)
    {
        $value = is_array($value) ?
            array_map('stripslashes_deep', $value) :
            stripslashes($value);

        return $value;
    }

    $_POST = array_map('stripslashes_deep', $_POST);
    $_GET = array_map('stripslashes_deep', $_GET);
    $_COOKIE = array_map('stripslashes_deep', $_COOKIE);
}
?>
```

6.11 Masquer PHP

En général, la sécurité par l'obscurité est une des formes de sécurité les plus faibles. Mais dans certains cas, chaque action, aussi faible soit elle, concernant la sécurité, est souhaitable.

Quelques astuces permettent de masquer PHP, et cela entrave les pirates qui recherchent des faiblesses dans votre système. En l'option `expose_php` à `off` dans votre fichier `php.ini`, vous pouvez réduire la quantité d'informations disponible.

Une autre astuce est de configurer le serveur web, comme Apache, pour qu'il utilise plusieurs types de fichiers différents avec PHP, soit localement avec le fichier `.htaccess`, soit dans le fichier de configuration lui-même. Vous pouvez utiliser des informations déroutantes comme ceci :

Masquer PHP avec un autre langage
--

<pre># Faire que le code PHP ressemble à un autre langage AddType application/x-httpd-php .asp .py .pl</pre>
--

Ou masquez le complètement :

Masquer PHP avec des types inconnus
--

<pre># Faire que le code PHP ressemble à un autre langage qui n'existe pas AddType application/x-httpd-php .bop .foo .133t</pre>
--

Ou encore, cachez-le sous forme de html. Cela a un léger impact négatif sur les performances générales, car tous les fichiers HTML seront aussi analysés et traités par le moteur PHP :

Utiliser le type html pour les extensions PHP
--

<pre># Faire que le code PHP ressemble à du html AddType application/x-httpd-php .htm .html</pre>

Pour que cela fonctionne efficacement, pensez à renommer tous vos fichiers avec les extensions ci-dessus. Même si c'est une forme de sécurité du non-dit, c'est une mesure de prévention mineure, avec peu d'inconvénients.

6.12 Etre à jour

PHP, comme de nombreux systèmes de grande taille, est constamment testé et amélioré. Chaque nouvelle version rassemble des modifications majeures et mineures, aussi bien pour renforcer la sécurité, que pour réparer les problèmes de conceptions de configuration, et d'autres points qui peuvent affecter la sécurité globale et la stabilité de votre système.

Comme les autres langages de scripts systèmes, la meilleure approche est de mettre à jour souvent PHP, et de rester à l'écoute des dernières versions et des modifications qu'elles apportent.

7 Caractéristiques

7.1 Identification HTTP avec PHP

Les fonctions d'identification HTTP de PHP ne sont disponibles que si PHP est exécuté comme module Apache, et non pas sous la forme d'un CGI. Sous cette forme, il est possible d'utiliser la fonction `header` pour demander une identification ("Authentication Required") au client, générant ainsi l'apparition d'une fenêtre de demande d'utilisateur et de mot de passe. Une fois que les champs ont été remplis, l'URL sera de nouveau appelée, avec les variables `PHP_AUTH_USER` , `PHP_AUTH_PW` et `AUTH_TYPE` contenant respectivement le nom d'utilisateur, le mot de passe et le type d'identification. Ces variables prédéfinies sont trouvées dans les tableaux `$_SERVER` et `$HTTP_SERVER_VARS` . Les méthodes d'identification simple ("Basic") et de type "Digest" (depuis PHP 5.1.0) sont supportées. Reportez-vous à la fonction `header` pour plus d'informations.

Note

Note sur les versions de PHP

Les auto-globales , telles que `$_SERVER` , ont été ajoutée en PHP à partir de la version 4.1.0 . `$HTTP_SERVER_VARS` est disponible depuis PHP 3.

Voici un exemple de script qui force l'identification du client pour accéder à une page :

Exemple d'identification HTTP simple

```
<?php
if (!isset($_SERVER['PHP_AUTH_USER'])) {
    header('WWW-Authenticate: Basic realm="My Realm"');
    header('HTTP/1.0 401 Unauthorized');
    echo 'Texte utilisé si le visiteur utilise le bouton d\'annulation';
    exit;
} else {
    echo "<p>Bonjour, {$_SERVER['PHP_AUTH_USER']}</p>";
    echo "<p>Votre mot de passe est {$_SERVER['PHP_AUTH_PW']}</p>";
}
?>
```

Exemple d'identification HTTP Digest

Cet exemple montre comment appliquer l'utilisation d'un script d'identification HTTP de type "Digest". Pour plus d'information, lisez la documentation [RFC 2617](#) .

```
<?php
$realm = 'Restricted area';

//utilisateur => mot de passe
$users = array('admin' => 'mypass', 'guest' => 'guest');

if (empty($_SERVER['PHP_AUTH_DIGEST'])) {
    header('HTTP/1.1 401 Unauthorized');
    header('WWW-Authenticate: Digest realm="'.$realm.
        '" qop="auth" nonce="'.uniqid().' " opaque="'.md5($realm).'");

    die('Texte utilisé si le visiteur utilise le bouton d\'annulation');
}

// analyse la variable PHP_AUTH_DIGEST
if (!( $data = http_digest_parse($_SERVER['PHP_AUTH_DIGEST']) ) ) ||
```

```

    !isset($users[$data['username']]))
    die('Mauvaise Pièces d\'identité!');

// Génération de réponse valide
$A1 = md5($data['username'] . ':' . $realm . ':' . $users[$data['username']]);
$A2 = md5($_SERVER['REQUEST_METHOD'] . ':' . $data['uri']);
$valid_response = md5($A1 . ':' . $data['nonce'] . ':' . $data['nc'] . ':' . $data['nonce'] . ':' . $data['qop']

if ($data['response'] != $valid_response)
    die('Mauvaise Pièces d\'identité!');

// ok, utilisateur & mot de passe valide
echo 'Vous êtes identifié en tant que : ' . $data['username'];

// fonction pour analyser l'en-tête http auth
function http_digest_parse($txt)
{
    // protection contre les données manquantes
    $needed_parts = array('nonce'=>1, 'nc'=>1, 'cnonce'=>1, 'qop'=>1, 'username'=>1, 'uri'=>1, 'r

    $data = array();

    preg_match_all('@(\w+)=(["?]) ([a-zA-Z0-9=.\/_-]+)\2@', $txt, $matches, PREG_SET_ORDER);

    foreach ($matches as $m) {
        $data[$m[1]] = $m[3];
        unset($needed_parts[$m[1]]);
    }

    return $needed_parts ? false : $data;
}
?>

```

Note**Note de compatibilité**

Soyez bien prudent lorsque vous utilisez des en-têtes HTTP avec PHP. Afin de garantir un maximum de compatibilité entre les navigateurs, le mot clé "Basic" doit être écrit avec un B majuscule, et le texte de présentation doit être placé entre guillemets simples, et exactement un espace doit précéder le code **401** dans l'en-tête **HTTP/1.0 401** .

Au lieu d'afficher simplement les variables globales PHP_AUTH_USER et PHP_AUTH_PW , vous préférerez sûrement vérifier la validité du nom d'utilisateur et du mot de passe. Par exemple, en envoyant ces informations à une base de données, ou en recherchant dans un fichier dbm.

Méfiez-vous des navigateurs bogués, tels que Internet Explorer. Ils semblent très susceptibles concernant l'ordre des en-têtes. Envoyer l'en-tête d'identification (WWW-Authenticate) avant le code de HTTP/1.0 401 semble lui convenir jusqu'à présent.

Pour éviter que quelqu'un écrive un script qui révèle les mots de passe d'une page, à la quelle on a accédé par une identification traditionnelle, les variables globales PHP_AUTH ne seront pas assignées si l'identification externe a été activée pour cette page. Dans ce cas, la variable REMOTE_USER peut être utilisée pour identifier l'utilisateur à l'extérieur. De même que \$_SERVER['REMOTE_USER'] .

Note**Note de configuration**

PHP utilise la présence de la directive AuthType pour déterminer si une identification externe est activée. Évitez donc cette directive de contexte si vous voulez utiliser l'identification de PHP (sinon,

les deux identifications se contrediraient, et échoueraient).

Notez cependant que les manipulations ci-dessus n'empêchent pas quiconque possède une page non identifiée de voler les mots de passe des pages protégées, sur le même serveur.

Netscape et Internet Explorer effaceront le cache d'identification client s'ils reçoivent une réponse 401. Cela permet de déconnecter un utilisateur, pour le forcer à saisir à nouveau son nom de compte et son mot de passe. Certains programmeurs l'utilisent pour donner un délai d'expiration, ou alors, fournissent un bouton de déconnexion.

Identification HTTP avec nom d'utilisateur/mot de passe forcé

```
<?php
function authenticate() {
    header('WWW-Authenticate: Basic realm="Test Authentication System"');
    header('HTTP/1.0 401 Unauthorized');
    echo "Vous devez entrer un identifiant et un mot de passe valide pour accéder
à cette ressource.\n";
    exit;
}

if ( !isset($_SERVER['PHP_AUTH_USER']) ||
    ($_POST['SeenBefore'] == 1 && $_POST['OldAuth'] == $_SERVER['PHP_AUTH_USER'])) {
    authenticate();
} else {
    echo "<p>Bienvenue : {$_SERVER['PHP_AUTH_USER']}<br />";
    echo "Ancien : {$_REQUEST['OldAuth']}";
    echo "<form action=\"{$_SERVER['PHP_SELF']}\" method=\"post\">\n";
    echo "<input type=\"hidden\" name=\"SeenBefore\" value=\"1\">\n";
    echo "<input type=\"hidden\" name=\"OldAuth\" value=\"{$_SERVER['PHP_AUTH_USER']}\">\n";
    echo "<input type=\"submit\" value=\"Identification\">\n";
    echo "</form></p>\n";
}
?>
```

Ce comportement n'est pas nécessaire par le standard d'identification HTTP Basic . Les tests avec Lynx ont montré qu'il n'affectait pas les informations de session lors de la réception d'un message de type 401. Ce qui fait que presser la touche "retour" (back) à un client Lynx précédemment identifié donnera l'accès direct à la ressource. Cependant, l'utilisateur peut utiliser la touche '_' pour détruire les anciennes identifications.

Notez également que avant PHP 4.3.3, l'identification HTTP ne fonctionne pas sous le serveur Microsoft IIS avec la version CGI de PHP dû à une limitation de IIS. Pour que cela fonctionne en PHP 4.3.3+, vous devez éditer votre configuration de IIS " Directory Security ". Cliquez sur " Edit " et activez uniquement " Anonymous Access ", tous les autres champs doivent être laissés non actifs.

Une autre limitation si vous utilisez le module IIS (ISAPI) et PHP 4, vous ne devez pas utiliser les variables PHP_AUTH_* mais à la place, la variable HTTP_AUTHORIZATION est disponible. Par exemple, utilisez le code suivant : `list($user, $pw) = explode(':', base64_decode(substr($_SERVER['HTTP_AUTHORIZATION'], 6)));`

Note

Note pour les utilisateurs de IIS :

Pour que l'identification HTTP fonctionne avec IIS, la directive PHP `cgi.rfc2616_headers` doit être définie à 0 (la valeur par défaut).

Note

Si le [safe mode](#) est activé, le uid de ce script est ajouté à la partie realm des en-têtes WWW-Authenticate .

7.2 Cookies

PHP supporte les cookies HTTP de manière transparente. Les cookies sont un mécanisme d'enregistrement d'informations sur le client, et de lecture de ces informations. Ce système permet d'identifier et de suivre les visiteurs. Vous pouvez envoyer un cookie avec la fonction [setcookie](#) ou [setrawcookie](#) . Les cookies font partie des en-têtes HTTP , ce qui impose que [setcookie](#) soit appelée avant tout affichage de texte. Ce sont les mêmes limitations que pour [header](#) . Vous pouvez utiliser les fonctions [de bufferisation de sortie](#) pour retarder l'affichage de votre script tant que vous n'avez pas décidé d'envoyer un cookie ou des en-têtes.

Tous les cookies qui sont envoyés au client seront automatiquement inclus dans un tableau auto-global `$_COOKIE` si [variables_order](#) contient "C". Si vous souhaitez affecter plusieurs valeurs à un seul cookie, ajoutez [] au nom du cookie.

Dépendamment de [register_globals](#) , les variables PHP régulières peuvent être créées à partir des cookies. Cependant, ce n'est pas recommandé de se fier sur elles puisque cette fonctionnalité est souvent désactivée pour des raisons de sécurité. `$HTTP_COOKIE_VARS` est aussi prédéfini dans les versions de PHP plus anciennes, lorsque la directive [track_vars](#) est activée. (Cette option est activée par défaut depuis PHP 4.0.3.)

Pour plus de détails, y compris des notes sur les bogues des navigateurs, voir les fonctions [setcookie](#) et [setrawcookie](#) .

7.3 Sessions

Le support des sessions de PHP est un mécanisme pour conserver des données entre différents accès. Cela vous permet de personnaliser les applications, et d'augmenter l'attrait de votre site. Toutes les informations sur les sessions sont disponibles dans la section [sessions](#) .

7.4 Utiliser les XForms

[XForms](#) est une variation des formulaires web traditionnels, qui permet leur utilisation sur diverses plates-formes et navigateurs, et même sur des médias moins traditionnels comme les documents PDF.

La première différence des XForms est leur présentation au client. [XForms for HTML Authors](#) contient une description détaillée de la création des XForms, complémentaire de notre tutoriel : nous allons nous consacrer à un exemple simple.

Un simple formulaire XForms de recherche

```
<h:html xmlns:h="http://www.w3.org/1999/xhtml"
        xmlns="http://www.w3.org/2002/xforms">
<h:head>
  <h:title>Search</h:title>
  <model>
    <submission action="http://example.com/search"
                method="post" id="s"/>
  </model>
</h:head>
```

```

<h:body>
  <h:p>
    <input ref="q"><label>Find</label></input>
    <submit submission="s"><label>Go</label></submit>
  </h:p>
</h:body>
</h:html>

```

Le formulaire ci-dessus affiche une boîte de texte (appelée q), et un bouton de soumission. Lorsque ce bouton est utilisé, le formulaire est envoyé à la page d' action .

C'est là que la différence se fait sentir, du point de vue de l'application web. Dans un formulaire HTML, les données sont envoyées au format application/x-www-form-urlencoded . Pour un XForm, c'est en fait un format XML qui est utilisé.

Si vous avez décidé d'utiliser des XForms, vous attendez peut être des données au format XML , et dans ce cas, regardez dans la variable \$HTTP_RAW_POST_DATA , où vous trouverez le document XML généré par le navigateur, que vous pourrez passer à votre moteur XSLT favori.

Si vous n'êtes pas intéressés par ce format, et que vous voulez simplement exploiter vos données avec la variable \$_POST , vous pouvez demander au navigateur de les envoyer au format application/x-www-form-urlencoded , en modifiant l'attribut de méthode method et en lui donnant la valeur de urlencoded-post .

Utiliser des XForms pour remplir \$_POST

```

<h:html xmlns:h="http://www.w3.org/1999/xhtml"
        xmlns="http://www.w3.org/2002/xforms">
<h:head>
  <h:title>Search</h:title>
  <model>
    <submission action="http://example.com/search"
                method="urlencoded-post" id="s"/>
  </model>
</h:head>
<h:body>
  <h:p>
    <input ref="q"><label>Find</label></input>
    <submit submission="s"><label>Go</label></submit>
  </h:p>
</h:body>
</h:html>

```

Note

Au moment de la rédaction de cet article, de nombreux navigateurs ne supportent pas les XForms. Vérifiez la version de votre navigateur si les exemples de ce tutoriel ne fonctionnent pas.

7.5 Gestion des chargements de fichier

7.5.1 Chargements de fichiers par méthode POST

Cette fonctionnalité permet aux personnes de télécharger à la fois du texte et des fichiers binaires. Avec les fonctions d'identifications et de manipulation de fichiers de PHP, vous avez le contrôle total pour définir qui a le droit de télécharger mais aussi ce qui sera fait du fichier une fois qu'il sera téléchargé.

PHP est capable de recevoir des fichiers émis par un navigateur conforme à la norme RFC-1867 (c'est-à-dire Netscape Navigator 3 ou supérieur, Microsoft Internet Explorer 3 avec un patch de Microsoft, ou supérieur sans le patch).

Note

Notes de configuration

Voir aussi les directives [file_uploads](#) , [upload_max_filesize](#) , [upload_tmp_dir](#) , [post_max_size](#) et [max_input_time](#) dans `php.ini`

PHP supporte aussi le chargement par la méthode PUT comme dans le navigateur Netscape Composer et Amaya du W3C. Reportez-vous au chapitre sur le [support de la méthode PUT](#) .

Formulaire de chargement de fichier

Un formulaire de téléchargement de fichiers peut être construit en créant un formulaire spécifique comme ceci :

```
<form enctype="multipart/form-data" action="_URL_" method="post">
  <input type="hidden" name="MAX_FILE_SIZE" value="30000" />
  Envoyez ce fichier : <input name="userfile" type="file" />
  <input type="submit" value="Envoyer le fichier" />
</form>
```

`_URL_` dans l'exemple précédent doit être remplacé et pointé vers un fichier PHP.

Le champs caché `MAX_FILE_SIZE` (mesuré en octets) doit précéder le champ input de type file et sa valeur représente la taille maximale acceptée du fichier. Ceci est considéré comme un conseil pour le navigateur, bien que PHP l'utilise également. Il est très facile de contourner cette restriction. Ne comptez pas sur le respect de cette configuration par le navigateur! La configuration de PHP sur la taille maximale à respecter ([upload_max_filesize](#)) ne peut être contournée, elle. Vous devez ajouter la variable `MAX_FILE_SIZE` à votre formulaire dans tous les cas car il prévient le chargement de gros fichiers qui demanderait un long délai d'attente au client et ainsi fera échouer le script.

Note

Assurez-vous que votre formulaire de téléchargement de fichier contienne `enctype="multipart/form-data"` sinon, le fichier se sera pas téléchargé.

La variable globale `$_FILES` existe depuis PHP 4.1.0 (Utilisez `$HTTP_POST_FILES` si vous utilisez une version plus ancienne). Ce tableau devrait contenir toutes les informations du fichier téléchargé.

Le contenu du tableau `$_FILES` est détaillé dans notre exemple ci-dessous. Notez que l'on suppose que le nom de la variable du fichier téléchargé est `userfile` , tel que défini dans le formulaire ci-dessus, mais peut être n'importe quel nom.

`$_FILES['userfile']['name']`

Le nom original du fichier, tel que sur la machine du client web.

`$_FILES['userfile']['type']`

Le type MIME du fichier, si le navigateur a fourni cette information. Par exemple, cela pourra être "image/gif" . Ce type mime n'est cependant pas vérifié du côté de PHP et donc, ne prend pas sa valeur pour se synchroniser.

`$_FILES['userfile']['size']`

La taille, en octets, du fichier téléchargé.

`$_FILES['userfile']['tmp_name']`

Le nom temporaire du fichier qui sera chargé sur la machine serveur.

`$_FILES['userfile']['error']`

Le code d'erreur [error code](#) associé au téléchargement de fichier. Cet élément a été introduit en PHP 4.2.0

Le fichier téléchargé sera stocké temporairement dans le dossier temporaire du système, à moins qu'un autre dossier soit fourni avec la directive `upload_tmp_dir` du `php.ini`. Le dossier par défaut du serveur peut être changé dans l'environnement via la variable `TMPDIR`. Modifier la valeur de cette variable avec la fonction `putenv` dans un script PHP sera sans effet. Cette variable d'environnement peut aussi être utilisée pour s'assurer que d'autres opérations fonctionnent aussi sur les fichiers téléchargés.

Validation de téléchargement de fichiers

Voyez aussi les fonctions `is_uploaded_file` et `move_uploaded_file` pour plus d'informations. L'exemple suivant va télécharger un fichier venant d'un formulaire.

```
<?php
// En PHP < 4.1.0, $HTTP_POST_FILES doit être utilisé
//   à la place de $_FILES.

$uploaddir = '/var/www/uploads/';
$uploadfile = $uploaddir . basename($_FILES['userfile']['name']);

echo '<pre>';
if (move_uploaded_file($_FILES['userfile']['tmp_name'], $uploadfile)) {
    echo "Le fichier est valide, et a été téléchargé
        avec succès. Voici plus d'informations :\n";
} else {
    echo "Attaque potentielle par téléchargement de fichiers.
        Voici plus d'informations :\n";
}

echo 'Voici quelques informations de débogage :';
print_r($_FILES);

echo '</pre>';

?>
```

Le script PHP qui reçoit le fichier chargé doit pouvoir gérer le fichier de manière appropriée. Vous pouvez utiliser la variable `$_FILES['userfile']['size']` pour recalculer tous les fichiers qui sont trop gros ou trop petits. Vous pouvez utiliser la variable `$_FILES['userfile']['type']` pour écarter les fichiers qui n'ont pas le bon type, mais l'utiliser uniquement pour une série de vérifications, car cette valeur est complètement sous le contrôle du client et n'est pas vérifiée du côté de PHP. Depuis PHP 4.2.0, vous pouvez utiliser l'information dans `$_FILES['userfile']['error']` et adapter votre politique en fonction des code d'erreur. Quelque soit votre politique, vous devriez soit effacer le fichier du dossier temporaire, soit le déplacer.

Si aucun fichier n'est sélectionné dans le formulaire, PHP retournera 0 dans `$_FILES['userfile']['size']` et rien du tout dans `$_FILES['userfile']['tmp_name']`.

Le fichier sera automatiquement effacé du fichier temporaire à la fin du script, s'il n'a pas été déplacé ou renommé.

Envoie d'un tableau de fichiers

PHP supporte les tableaux en HTML ainsi qu'avec les fichiers.

```
<form action="" method="post" enctype="multipart/form-data">
<p>Images:
<input type="file" name="pictures[]" />
<input type="file" name="pictures[]" />
<input type="file" name="pictures[]" />
<input type="submit" value="Send" />
</p>
```

```

</form>

<?php
foreach ($_FILES["pictures"]["error"] as $key => $error) {
    if ($error == UPLOAD_ERR_OK) {
        $tmp_name = $_FILES["pictures"]["tmp_name"][$key];
        $name = $_FILES["pictures"]["name"][$key];
        move_uploaded_file($tmp_name, "data/$name");
    }
}
?>

```

7.5.2 Explication sur les messages d'erreurs de chargement de fichiers

Depuis PHP 4.2.0, PHP retourne un code d'erreur approprié dans le tableau de fichiers. Ce code d'erreur est accessible à l'index ['error'] du tableau, qui est créé durant le téléchargement, par PHP. En d'autres termes, le message d'erreur est accessible dans la variable `$_FILES['userfile']['error']`.

UPLOAD_ERR_OK

Valeur : 0. Aucune erreur, le téléchargement est correct.

UPLOAD_ERR_INI_SIZE

Valeur : 1. Le fichier téléchargé excède la taille de `upload_max_filesize`, configuré dans le `php.ini`.

UPLOAD_ERR_FORM_SIZE

Valeur : 2. Le fichier téléchargé excède la taille de **MAX_FILE_SIZE**, qui a été spécifiée dans le formulaire HTML.

UPLOAD_ERR_PARTIAL

Valeur : 3. Le fichier n'a été que partiellement téléchargé.

UPLOAD_ERR_NO_FILE

Valeur : 4. Aucun fichier n'a été téléchargé.

UPLOAD_ERR_NO_TMP_DIR

Valeur : 6. Un dossier temporaire est manquant. Introduit en PHP 4.3.10 et PHP 5.0.3.

UPLOAD_ERR_CANT_WRITE

Valeur : 7. Échec de l'écriture du fichier sur le disque. Introduit en PHP 5.1.0.

Note

Ces constantes sont apparues en PHP 4.3.0.

7.5.3 Erreurs classiques

La variable `MAX_FILE_SIZE` ne peut pas spécifier une taille de fichier plus grande que la taille qui a été fixée par `upload_max_filesize`, dans le `php.ini`. La valeur par défaut est 2 megaoctets.

Si une limite de mémoire est activée, une plus grande valeur de `memory_limit` peut être nécessaire. Assurez-vous d'avoir défini une valeur pour `memory_limit` assez grande.

Si la valeur de `max_execution_time` est trop petite, le temps d'exécution du script peut excéder cette valeur. Assurez-vous d'avoir défini une valeur pour `max_execution_time` assez grande.

Note

`max_execution_time` affecte uniquement le temps d'exécution du script. Le temps passé sur l'activité qui apparaît en dehors de l'exécution du script comme les appels systèmes avec la fonction `system`, la fonction `sleep`, les requêtes sur les bases de données, le temps mis pour

effectuer le téléchargement du fichier, etc. n'est pas inclus lors du calcul du temps maximal de l'exécution du script.

Attention

`max_input_time` définit le temps maximal, en secondes, au script pour recevoir les données ; cela inclut le téléchargement du fichier. Pour les fichiers multiples, ou les gros fichiers, ou encore pour les utilisateurs sur des connexions lentes, la valeur par défaut de 60 secondes peut être dépassée.

Ne pas valider les fichiers que vous manipulez peut donner l'accès aux utilisateurs à des fichiers sensibles dans d'autres dossiers!

Attention : il semble que CERN httpd supprime tout ce qui est après le premier caractère dans l'entête MIME. Tant que c'est le cas, CERN httpd ne pourra pas effectuer de chargements de fichiers.

Du fait de la grande diversité des systèmes, nous ne pouvons garantir que les fichiers avec des noms exotiques (comme, par exemple, contenant des espaces) seront traités correctement.

Le développeur ne doit pas mixer les champs input normaux et les champs input de téléchargement dans une même variable (en utilisant un nom d'input comme `foo[]`).

7.5.4 Télécharger plusieurs fichiers simultanément

Le téléchargement de plusieurs fichiers est possible en utilisant des noms différents dans l'attribut `name` de la balise `input`.

Il est aussi possible de télécharger plusieurs fichiers simultanément et d'obtenir les informations sous forme de tableau. Pour cela, vous devez utiliser la syntaxe de tableau dans les noms de balises HTML, comme vous l'avez fait avec les sélections multiples et les boîtes à cocher.

Note

Le support des téléchargements multiples a été ajouté en PHP 3.0.10.

Télécharger plusieurs fichiers simultanément

```
<form action="file-upload.php" method="post" enctype="multipart/form-data">
  Envoyez plusieurs fichiers : <br />
  <input name="userfile[]" type="file" /><br />
  <input name="userfile[]" type="file" /><br />
  <input type="submit" value="Envoyer les fichiers" />
</form>
```

Lorsque le formulaire ci-dessus a été envoyé, les tableaux `$_FILES['userfile']`, `$_FILES['userfile']['name']`, et `$_FILES['userfile']['size']` seront initialisés (tout comme `$HTTP_POST_FILES` pour les versions de PHP antérieure à la 4.1.0). Lorsque `register_globals` est activé, les variables globales concernant les fichiers téléchargés sont aussi initialisées. Chacune d'entre elle contiendra un tableau numériquement indexé, avec les valeurs décrivant les fichiers téléchargés.

Par exemple, supposons que les fichiers `/home/test/review.html` et `/home/test/xwp.out` ont été téléchargés. Dans ce cas, `$_FILES['userfile']['name'][0]` contient `review.html` et `$_FILES['userfile']['name'][1]` contient `xwp.out`. Similairement, `$_FILES['userfile']['size'][0]` va contenir la taille du fichier `review.html`, etc.

`$_FILES['userfile']['name'][0]`, `$_FILES['userfile']['tmp_name'][0]`, `$_FILES['userfile']['size'][0]` et `$_FILES['userfile']['type'][0]` sont aussi créées.

7.5.5 Chargement par méthode PUT

Le support de la méthode PUT a été modifié entre PHP 3 et PHP 4. En PHP 4, il faut utiliser le flux d'entrée standard pour recevoir le contenu du fichier.

Sauver un fichier par méthode PUT en PHP 4

```
<?php
/* Les données via la méthode PUT arrivent dans le flux stdin */
$putdata = fopen("php://stdin", "r");

/* Ouvre un fichier pour écriture */
$fp = fopen("myputfile.ext", "w");

/* Lit les données par bloc de 1ko
   et les écrit dans le fichier */
while ($data = fread($putdata, 1024)) {
    fwrite($fp, $data);
}

/* Ferme les flux */
fclose($fp);
fclose($putdata);
?>
```

Note

Toute la documentation ci-dessous s'applique à PHP 3 uniquement.

PHP supporte la méthode HTTP PUT utilisée par les navigateurs tels que Netscape Composer et Amaya du W3C. Les requêtes de type PUT sont beaucoup plus simples que les chargements de fichiers, et elles ressemblent à :

Méthode PUT pour les chargements de fichiers

```
PUT /path/filename.html HTTP/1.1
```

Normalement, cela signifie que le client distant va sauver les données qui suivent dans le fichier : /path/filename.html de votre disque. Ce n'est évidemment pas très sécurisé de laisser Apache ou PHP écraser n'importe quel fichier de l'arborescence. Pour éviter ceci, il faut d'abord dire au serveur que vous voulez qu'un script PHP donné gère la requête. Avec Apache, il y a une directive pour cela : **Script** . Elle peut être placée n'importe où dans le fichier de configuration d'Apache. En général, les webmasters la place dans le bloc <Directory> , ou peut être dans le bloc <Virtualhost> . La ligne suivante fera très bien l'affaire :

Directive Apache pour le chargement par méthode PUT

```
Script PUT /put.php
```

Elle indique à Apache qu'il doit envoyer les requêtes de chargement par méthode PUT au script put.php . Bien entendu, cela présuppose que vous avez activé PHP pour qu'il prenne en charge les fichiers de type .php , et que PHP est actif.

Dans le fichier put.php , vous pouvez mettre ceci :

Réception d'un fichier par méthode PUT

```
<?php
copy($PHP_UPLOADED_FILE_NAME, $DOCUMENT_ROOT . $REQUEST_URI);
?>
```

Ce script va copier le fichier chargé par le client distant à l'endroit désiré. Vous aurez probablement à effectuer quelques tests et l'identification de l'utilisateur, avant d'effectuer cette copie. Le seul piège est que lorsque PHP reçoit un chargement par méthode PUT, il va enregistrer le fichier dans

le dossier temporaire, tout comme avec la méthode POST . A la fin de la requête, le fichier sera effacé. Ce qui fait que ce script doit placer le fichier chargé quelque part. Le nom du fichier temporaire est placé dans la variable globale \$PHP_PUT_FILENAME , et la destination prévue est placée dans \$REQUEST_URI (ces noms peuvent changer d'une configuration d'Apache à l'autre). Cette destination est celle qui est demandée par le client, et vous n'avez pas à obéir aveuglément au client. Vous pourriez par exemple, déplacer le fichier dans un dossier de chargement.

7.6 Utilisation des fichiers à distance

Aussi longtemps que le support des gestionnaires d'URL ("URL fopen wrapper") sont activés dans le php.ini , avec l'option allow_url_fopen , vous pouvez utiliser des URLs (HTTP et FTP) avec la majorité des fonctions qui utilisent un nom de fichier comme paramètre. Cela inclut notamment include , include_once , require et require_once . Reportez-vous à Liste des protocoles supportés pour plus d'informations sur les protocoles supportés par PHP.

Note

En PHP 4.0.3 et plus récent, pour pouvoir utiliser les gestionnaires d'URL, vous devez configurer PHP avec l'option --enable-url-fopen-wrapper .

Note

Les versions de PHP pour Windows plus anciennes que 4.3 de supportait pas les gestionnaires d'URL avec les fonctions suivantes : include , include_once , require , require_once , et les fonctions de création d'images de l'extension Images .

Par exemple, vous pouvez suivre l'exemple suivant pour ouvrir un fichier sur un serveur web distant, analyser les résultats pour extraire les informations dont vous avez besoin, et ensuite l'utiliser dans une requête de base de données, ou simplement éditer les informations dans le style de votre site.

Connaître le titre d'une page distante

```
<?php
$file = fopen ("http://www.example.com/", "r");
if (!$file) {
    echo "<p>Impossible de lire la page.\n";
    exit;
}
while (!feof ($file)) {
    $line = fgets ($file, 1024);
    /* Cela ne fonctionne que si les balises Title sont correctement utilisées */
    if (eregi("<title>(.*?)</title>", $line, $out)) {
        $title = $out[1];
        break;
    }
}
fclose($file);
?>
```

Vous pouvez aussi écrire des fichiers sur un serveur FTP aussi longtemps que vous êtes connecté avec un utilisateur ayant les bons droits d'accès, alors que le fichier n'existait pas encore.

Pour vous connecter avec un utilisateur autre qu'anonyme, vous devez spécifier un nom d'utilisateur (et certainement le mot de passe) dans l'URL, comme ftp://user:password@ftp.example.com/path/to/file . (Vous pouvez utiliser le même type de syntaxe pour accéder aux fichiers via HTTP lorsqu'ils nécessitent une identification simple).

Stocker des données sur un serveur distant

```

<?php
$file = fopen ("ftp://ftp.example.com/incoming/outputfile", "w");
if (!$file) {
    echo "<p>Unable to open remote file for writing.\n";
    exit;
}
/* Ecriture des données. */
fputs ($file, $_SERVER['HTTP_USER_AGENT'] . "\n");
fclose ($file);
?>

```

Note

Remarque : Vous pouvez avoir l'idée, à partir de l'exemple ci-dessus, d'utiliser la même technique pour écrire sur un log distant, mais comme mentionné ci-dessus vous ne pouvez qu'écrire sur un nouveau fichier en utilisant les fonctions fopen avec une URL. Pour faire des log distribués, nous vous conseillons de regarder la partie syslog .

7.7 Gestion des connexions

Note

Les informations suivantes ne sont valables qu'à partir de la version 3.0.7.

Le statut des connexions est conservé en interne par PHP. Il y a trois états possibles :

- 0 - NORMAL (normal)
- 1 - ABORTED (annulé)
- 2 - TIMEOUT (périmé)

Lorsqu'un script PHP est en cours d'exécution, son état est NORMAL. Si le client distant se déconnecte, le statut devient ABORTED. En général, une telle déconnexion provient d'un arrêt temporaire. Si la durée maximale d'exécution de PHP est dépassée, (voir set_time_limit), le script prend le statut TIMEOUT.

Vous pouvez en outre décider si vous voulez que la déconnexion d'un client provoque l'arrêt de votre script. Il est parfois pratique que vos scripts continuent à s'exécuter jusqu'à la fin, même si le client n'est plus là pour recevoir les informations. Cependant, par défaut, le script s'arrêtera dès que le client se déconnecte. Ce comportement peut être modifié avec la directive ignore_user_abort dans le fichier php.ini ou bien avec la directive Apache ignore_user_abort du fichier Apache httpd.conf ou avec la fonction ignore_user_abort . Si vous ne demandez pas à PHP d'ignorer la déconnexion, et que l'utilisateur se déconnecte, le script sera terminé. La seule exception est si vous avez enregistré une fonction de fermeture, avec register_shutdown_function . Avec une telle fonction, lorsque l'utilisateur interrompt sa requête, à la prochaine exécution du script, PHP va s'apercevoir que le dernier script n'a pas été terminé, et il va déclencher la fonction de fermeture. Cette fonction sera aussi appelée à la fin du script, si celui-ci se termine normalement. Pour pouvoir avoir un comportement différent suivant l'état du script lors de sa finalisation, vous pouvez exécuter des commandes spécifiques à la déconnexion grâce à la commande connection_aborted . Cette fonction retournera TRUE si la connexion a été annulée.

Votre script peut aussi être automatiquement interrompu après une certaine durée de temps. Par défaut, le délai est de 30 secondes. Cette valeur peut être changée en utilisant la directive PHP max_execution_time dans le fichier php.ini ou avec la directive php_value max_execution_time , dans le fichier Apache httpd.conf ou encore avec la fonction set_time_limit . Lorsque le délai expire, le script est terminé, et comme pour la déconnexion du client, une fonction de finalisation sera appelée. Dans cette fonction, vous pouvez savoir si c'est le délai d'expiration qui a causé la fin du script, en appelant la fonction connection_status . Cette fonction retournera 2 si le délai d'expiration a été dépassé.

Une chose à noter est que les deux cas ABORTED et TIMEOUT peuvent être appelés en même temps. Ceci est possible si vous demandez à PHP d'ignorer les déconnexions des utilisateurs. PHP va quand même noter le fait que l'utilisateur s'est déconnecté, mais le script va continuer. Puis, lorsqu'il atteint la limite de temps, le script va expirer. A ce moment-là, la fonction `connection_status` retournera 3 .

7.8 Connexions persistantes aux bases de données

Les connexions persistantes aux bases de données SQL sont des connexions qui ne se referment pas à la fin du script. Lorsqu'une connexion persistante est demandée, PHP s'assure qu'il n'y a pas une autre connexion identique (qui serait ouverte précédemment, avec le même nom d'hôte, d'utilisateur et le même mot de passe), et si une telle connexion existe, elle est utilisée. Sinon, elle est créée. Une connexion identique est une connexion qui a ouvert le même hôte, avec le même nom et le même mot de passe (s'ils sont nécessaires).

Ceux qui ne sont pas rompus aux techniques des serveurs web et leur distribution de la charge de travail, se font parfois une fausse idée de ces connexions persistantes. En particulier, les connexions persistantes **ne permettent pas** l'ouverture de plusieurs sessions avec le même lien; elles **ne permettent pas** la réalisation de transactions efficaces et ne font pas le café. En fait, pour être extrêmement clair sur le sujet, les connexions persistantes **ne vous donnent** aucune fonctionnalité de plus que les connexions non persistantes.

Alors pourquoi les connexions persistantes?

Cela s'explique par la manière avec laquelle les serveurs web fonctionnent. Il y a trois manières d'utiliser PHP pour générer des pages.

La première est d'utiliser PHP comme un CGI (Common Interface Gateway). Lorsque PHP fonctionne de cette manière, une instance de l'interpréteur PHP est créée puis détruite pour chaque page demandée. Etant donné que cet interpréteur est détruit après chaque requête, toutes les ressources acquises (comme une connexion à une base SQL), sont purement et simplement détruites.

La deuxième méthode, et de loin, la plus prisée, est d'exécuter PHP sous la forme d'un module sur un serveur multi-processus, ce qui revient à dire : Apache. Un tel serveur a typiquement un processus parent qui coordonne un ensemble de processus fils, qui servent les fichiers. Lorsque les requêtes parviennent depuis un client, elles sont transmises à un fils disponible. Cela signifie que si un client fait une deuxième requête, il peut être servi par un processus client différent du premier. Les connexions persistantes vous permettent alors de ne vous connecter à une base SQL que la première fois. Lors des connexions ultérieures, les processus fils pourront réutiliser la connexion ouverte précédemment.

La dernière méthode est d'utiliser PHP sous la forme d'un module de serveur multi-threads. Actuellement, PHP 4 supporte ISAPI, WSAPI, et NSAPI (sous Windows), qui permettent tous d'utiliser PHP comme un module sur un serveur multi-threads tel que Netscape FastTrack (iPlanet), Microsoft Internet Information Server (IIS), et O'Reilly's WebSite Pro. Le comportement est essentiellement le même que pour les serveurs multi-processus décrits précédemment. Notez que SAPI n'est pas disponible avec PHP 3.

Si les connexions persistantes n'ont aucune fonctionnalité de plus, à quoi servent-elles?

La réponse est extrêmement simple : efficacité. Les connexions persistantes sont un bon moyen d'accélérer les accès à une base SQL si le traitement de connexion à la base est long. Ce temps

dépend de nombreux facteurs : le type de base de données, cette base est-elle sur le même serveur ou pas, quelle est la charge du serveur de base de données, etc. Si le temps de connexion est long, les connexions persistantes seront bien utiles, car une fois ouverte par un processus fils, la connexion est réutilisable sans avoir à se reconnecter. Si vous avez 20 processus fils, il suffit d'avoir 20 connexions persistantes ouvertes, une par fils.

Notez que les connexions persistantes ont quelques inconvénients si vous hébergez une base de données, dont le nombre maximal de connexion risque d'être atteint par les connexions persistantes. Si votre base de données accepte jusqu'à 16 connexions simultanées et que, 17 processus essaient de se connecter, le dernier restera sur la touche. S'il y a des erreurs dans les scripts qui ne permettent pas de fermer la connexion (par exemple, une boucle infinie), votre serveur sera rapidement engorgé. Vérifiez la documentation de votre base de données pour savoir comment elle traite les connexions inactives ou abandonnées.

Attention

Il y a quelques autres limitations à bien garder en tête lorsque vous utilisez les connexions persistantes. L'une est que lorsque vous posez un verrou avec une connexion persistante, si le script ne peut libérer le verrou pour une raison quelconque, alors les autres scripts qui réutiliseront la connexion seront bloqué indéfiniment, et vous imposerez le redémarrage du serveur ou de la base de données. Une autre limitation est, lors de l'utilisation des transactions, un bloc de transaction non fermé sera prolongé au script suivant, s'il n'est pas fermé par le script initiateur. Dans les deux cas, vous pouvez utiliser la fonction `register_shutdown_function` pour enregistrer une fonction simple de nettoyage, pour déverrouiller les tables, et annuler les transactions en cours. Mieux encore, évitez le problème entièrement en n'utilisant pas les connexions persistantes dans les scripts qui ont besoin de verrous ou de transactions. Vous pourrez toujours les utiliser ailleurs.

Résumons-nous : les connexions persistantes ont été définies pour avoir les mêmes fonctionnalités que les connexions non persistantes. Les deux types de connexions sont **parfaitement interchangeables**, et n'affecteront pas le comportement de votre script : uniquement son efficacité.

Voir aussi [fbsql_pconnect](#), [ibase_pconnect](#), [ifx_pconnect](#), [ingres_pconnect](#), [msql_pconnect](#), [mssql_pconnect](#), [mysql_pconnect](#), [oci_plogon](#), [odbc_pconnect](#), [ora_plogon](#), [pfsockopen](#), [pg_pconnect](#) et [sybase_pconnect](#).

7.9 Safe mode

Le "Safe Mode" est le mode de sécurité de PHP : une solution au problème de partage de PHP sur un serveur. Ce système pêche au niveau de l'architecture car il n'est pas correct de tenter de résoudre ce problème au niveau de PHP, mais les solutions alternatives basées sur le serveur web et l'OS ne sont pas réalistes. De nombreux intervenants, notamment les fournisseurs d'hébergement, utilisent le "Safe Mode".

7.9.1 Sécurité et Safe Mode

Nom	Par défaut	Modifiable	Historique
<code>safe_mode</code>	"0"	PHP_INI_SYSTEM	
<code>safe_mode_gid</code>	"0"	PHP_INI_SYSTEM	Disponible depuis PHP 4.1.0.
<code>safe_mode_include_dir</code>	NULL	PHP_INI_SYSTEM	Disponible

			depuis PHP 4.1.0.
safe_mode_exec_dir	""	PHP_INI_SYSTEM	
safe_mode_allowed_env_vars	"PHP_"	PHP_INI_SYSTEM	
safe_mode_protected_env_vars	"LD_LIBRARY_PATH"	PHP_INI_SYSTEM	
open_basedir	NULL	PHP_INI_SYSTEM	
disable_functions	""	php.ini seulement	Disponible depuis PHP 4.0.1.
disable_classes	""	php.ini seulement	Disponible depuis PHP 4.3.2.

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [Directives du php.ini](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

safe_mode booléen

Active ou non le mode de sécurité de PHP.

safe_mode_gid booléen

Par défaut, le safe mode fait une comparaison des propriétaires, avant d'ouvrir un fichier. Si vous voulez alléger un peu ce niveau de sécurité, vous pouvez réaliser une comparaison de groupe, et activer cette directive. Si cette directive vaut FALSE (sa valeur par défaut), c'est une comparaison sur les UID , et sinon, sur les GID .

safe_mode_include_dir chaîne de caractères

Les vérifications basées sur le UID ou GID sont ignorées lorsque les fichiers inclus sont placés dans le dossier indiqué par cette directive, ainsi que ses sous-dossiers. Les dossiers peuvent être aussi dans l' include_path ou bien il faut inclure le chemin complet.

Depuis PHP 4.2.0, cette directive utilise le point virgule de la même façon que le fait include_path , pour permettre de configurer plusieurs dossiers.

La restriction spécifiée est en fait un préfixe, plus qu'un nom de dossier. Cela signifie que " safe_mode_include_dir = /dir/incl " autorise aussi bien " /dir/include " que " /dir/incls " , s'ils existent. Lorsque vous souhaitez restreindre l'accès à un dossier spécifique, il faut terminer cette directive avec un slash / . Par exemple " safe_mode_include_dir = /dir/incl/ " .

Si la valeur de cette directive est vide, aucun fichier avec le UID / GID différent ne peut être inclus dans PHP 4.2.3 et dans les versions PHP 4.3.3 et plus récentes. Dans les versions antérieures, tous les fichiers pouvaient être inclus.

safe_mode_exec_dir chaîne de caractères

Si PHP est utilisé en safe mode, les fonctions comme system et toutes celles qui permettent l'exécution en ligne de commande refuseront d'exécuter des programmes qui ne sont pas dans ce dossier. Vous devez utiliser / en tant que séparateur de dossier sous tous les environnements, y compris Windows.

safe_mode_allowed_env_vars chaîne de caractères

Modifier certaines variables d'environnement est un trou de sécurité potentiel. Cette directive contient une liste de noms variables d'environnement séparées par des virgules, ou de préfixes. En Safe mode, l'utilisateur ne pourra modifier que les variables d'environnement dont le nom commence par l'un des préfixes fourni ici. Par défaut, les utilisateurs ne peuvent modifier que les variables d'environnement qui commencent par PHP_ (e.g. PHP_FOO=BAR).

Note

Si cette directive est vide, PHP autorisera la modification de TOUTES les variables d'environnement.

`safe_mode_protected_env_vars` chaîne de caractères

Cette directive contient une liste de variables d'environnement que le programmeur ne pourra pas modifier en utilisant la fonction `putenv`. Ces variables seront protégées, même si la directive `safe_mode_allowed_env_vars` autorise leur modification.

`open_basedir` chaîne de caractères

Limite les fichiers accessibles par PHP dans l'arborescence. Cette directive **n'est pas** affectée par le safe mode.

Lorsqu'un script tente d'ouvrir un fichier, avec les fonctions `fopen` ou `gzopen`, la situation du fichier est vérifiée. Si le fichier se situe hors du dossier spécifié dans cette directive, PHP refusera de l'ouvrir. Les liens symboliques sont résolus, ce qui fait que cette restriction ne peut être contournée par un lien symbolique.

La valeur spéciale `.` indique que le dossier dans lequel le script est stocké, servira de dossier de base. Cela est cependant un peu dangereux car le dossier courant du script peut facilement être modifié avec la fonction `chdir`.

Dans `httpd.conf`, `open_basedir` peut être désactivée (i.e. pour certains hôtes virtuels) de la même manière que toute autre directive de configuration avec la syntaxe "`php_admin_value open_basedir none`".

Sous Windows, séparez les dossiers par des points virgules. Sur les autres systèmes, séparez les dossiers avec des deux-points. Lorsque PHP est utilisé comme module Apache, les chemins de la directive `open_basedir` des dossiers parents sont automatiquement transmis.

La restriction spécifiée par `open_basedir` est en fait un préfixe et non un dossier. Cela signifie que "`open_basedir = /dir/incl`" donne accès au dossier `/dir/include` et aussi au dossier `/dir/incls` s'il existe. Lorsque vous souhaitez restreindre l'accès à un dossier spécifique, ajoutez un slash final. Par exemple : `open_basedir = /dir/incl/`.

Note

Le support des dossiers multiples a été ajouté en PHP 3.0.7.

La valeur par défaut permet l'ouverture de tous les fichiers.

`disable_functions` chaîne de caractères

Cette directive vous permet de désactiver certaines fonctions pour des raisons de sécurité. Elle prend une liste de nom de fonctions, séparés par des virgules. `disable_functions` n'est pas affectée par le safe mode.

Cette directive doit être configurée dans le fichier `php.ini`. Par exemple, vous ne pourrez pas la configurer dans le fichier `httpd.conf`.

`disable_classes` chaîne de caractères

Cette directive vous permet de désactiver certaines classes pour des raisons de sécurité. Elle prend une liste de nom de fonctions, séparées par des virgules. `disable_functions` n'est pas affectée par le safe mode.

Cette directive doit être configurée dans le fichier `php.ini`. Par exemple, vous ne pourrez pas la configurer dans le fichier `httpd.conf`.

Note
Disponibilité
Cette directive est disponible depuis PHP 4.3.2

Voir aussi [register_globals](#) , [display_errors](#) et [log_errors](#) .

Lorsque [safe_mode](#) est actif, PHP vérifie que le propriétaire du script courant est le même que le propriétaire des fichiers ou dossiers qui seront manipulés par ce script. Par exemple, dans la situation suivante :

```
-rw-rw-r--  1 rasmus  rasmus      33 Jul  1 19:20 script.php
-rw-r--r--  1 root   root        1116 May 26 18:01 /etc/passwd
```

exécuter le script `script.php`

```
<?php
  readfile('/etc/passwd');
?>
```

générera cette erreur, si le Safe Mode est activé :

```
Warning: SAFE MODE Restriction in effect. The script whose uid is 500 is not
allowed to access /etc/passwd owned by uid 0 in /docroot/script.php on line 2
```

Cependant, il arrive que la vérification faite avec le nom du propriétaire du fichier soit trop restrictive, et qu'une vérification sur le nom du groupe soit suffisante. C'est une autre fonctionnalité supportée par la directive [safe_mode_gid](#) . En lui donnant la valeur de On , les vérifications seront faites sur le GID , alors qu'en lui laissant sa valeur par défaut de Off , les vérifications seront faites sur le UID .

Si vous utilisez la directive [open_basedir](#) au lieu du [safe_mode](#) , alors les manipulations seront limitées aux fichiers situés dans les dossiers spécifiés. Par exemple (Apache httpd.conf) :

```
<Directory /docroot>
  php_admin_value open_basedir /docroot
</Directory>
```

Si vous exécutez le script `script.php` ci-dessus avec la configuration d'[open_basedir](#), le résultat sera l'affichage suivant :

```
Warning: open_basedir restriction in effect. File is in wrong directory in
/docroot/script.php on line 2
```

Vous pouvez aussi désactiver individuellement les fonctions. Par exemple, en ajoutant cette ligne dans le fichier `php.ini` :

```
disable_functions readfile,system
```

toute utilisation des fonctions [readfile](#) et [system](#) générera l'affichage suivant :

```
Warning: readfile() has been disabled for security reasons in
/docroot/script.php on line 2
```

Attention
Ces restrictions de PHP sont bien sûr invalides en exécution binaire.

7.9.2 Fonctions désactivées par le Safe Mode

Voici une liste non-exhaustive des fonctions désactivées par le [Safe Mode](#) .

Fonction	Limitations
dbmopen	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
dbase_open	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
filepro	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
filepro_rowcount	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
filepro_retrieve	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
Fonctions ifx_*	restrictions sql_safe_mode, (!= Safe Mode)
Fonctions ingres_*	restrictions sql_safe_mode, (!= Safe Mode)
Fonctions mysql_*	restrictions sql_safe_mode, (!= Safe Mode)
pg_lo_import	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
posix_mkfifo	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
putenv	Obéis aux directives <code>safe_mode_protected_env_vars</code> et <code>safe_mode_allowed_env_vars</code> . Voir aussi la documentation de putenv
move_uploaded_file	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
chdir	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
dl	Cette fonction est désactivée par le safe-mode
opérateur guillemets obliques	Cette fonction est désactivée par le safe-mode
shell_exec (équivalent fonctionnel des guillemets obliques)	Cette fonction est désactivée par le safe-mode
exec	Vous ne pouvez exécuter que les programmes qui sont dans le dossier <code>safe_mode_exec_dir</code> . Pour des raisons pratiques, il n'est pas possible d'utiliser des jokers comme <code>..</code> dans le chemin de ce dossier. <code>escapeshellcmd</code> est exécuté sur les

	arguments de cette fonction.
<u>system</u>	Vous ne pouvez exécuter que les programmes qui sont dans le dossier <u>safe_mode_exec_dir</u> . Pour des raisons pratiques, il n'est pas possible d'utiliser des jokers comme .. dans le chemin de ce dossier. <u>escapeshellcmd</u> est exécuté sur les arguments de cette fonction.
<u>passthru</u>	Vous ne pouvez exécuter que les programmes qui sont dans le dossier <u>safe_mode_exec_dir</u> . Pour des raisons pratiques, il n'est pas possible d'utiliser des jokers comme .. dans le chemin de ce dossier. <u>escapeshellcmd</u> est exécuté sur les arguments de cette fonction.
<u>popen</u>	Vous ne pouvez exécuter que les programmes qui sont dans le dossier <u>safe_mode_exec_dir</u> . Pour des raisons pratiques, il n'est pas possible d'utiliser des jokers comme .. dans le chemin de ce dossier. <u>escapeshellcmd</u> est exécuté sur les arguments de cette fonction.
<u>fopen</u>	Note
	Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
<u>mkdir</u>	Note
	Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
<u>rmdir</u>	Note
	Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
<u>rename</u>	Note
	Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
	Note
	Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
<u>unlink</u>	Note
	Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
	Note
	Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
<u>copy</u>	Note
	Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
	Note
	Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. (sur source et target)
<u>chgrp</u>	Note
	Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.

<u>chown</u>	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
<u>chmod</u>	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. De plus, vous ne pouvez pas modifier les SUID, SGID et le bit sticky
<u>touch</u>	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
<u>symlink</u>	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. (note : seule l'hôte cible est vérifié)
<u>link</u>	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. (note : seule le fichier de destination est vérifié.)
<u>apache_request_headers</u>	En Safe Mode, les en-têtes commençants par authorization (sensible à la casse) ne seront pas retournés.
<u>header</u>	Avec le safe mode, le uid du script est ajouté à la partie realm de l'en-tête WWW-Authenticate si vous utilisez cet en-tête pour l'identification.
<u>variables PHP_AUTH</u>	Avec le safe mode, les variables PHP_AUTH_USER , PHP_AUTH_PW et PHP_AUTH_TYPE ne sont pas disponibles dans la variable \$_SERVER . Indépendamment, vous pouvez utiliser la variable REMOTE_USER pour connaître l'utilisateur. (note : affectée uniquement depuis PHP 4.3.0)
<u>highlight_file</u> , <u>show_source</u>	Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. Note Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. (note : affectée uniquement depuis PHP 4.2.1)
<u>parse_ini_file</u>	Note

	Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
	Note
	Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
	(note : affectée uniquement depuis PHP 4.2.1)
<u>set_time_limit</u>	N'a aucun effet lorsque PHP fonctionne avec le <u>safe mode</u> .
<u>max_execution_time</u>	N'a aucun effet lorsque PHP fonctionne avec le <u>safe mode</u> .
<u>mail</u>	Si le <u>safe mode</u> est actif, le 5ème paramètre est désactivé (note : uniquement affecté depuis PHP 4.2.3)
Toutes les fonctions qui utilisent les wrappers de fichiers. (php4/main/fopen_wrappers.c)	??

7.10 Utiliser PHP en ligne de commande

Depuis la version 4.3.0, PHP supporte un nouveau type de SAPI (Server Application Programming Interface , c'est à dire Interface de Programmation d'Applications Serveur) appelé CLI , ce qui signifie Command Line Interface et se traduit par Interface de Ligne de Commande . Comme son nom l'indique, ce type SAPI cible les applications shell (ou desktop), écrites en PHP. Il y a un certain nombre de différences entre le type CLI SAPI et les autres SAPI expliqués dans ce chapitre. Il convient de préciser que CLI et CGI sont des SAPI différentes, bien qu'ils partagent des comportements similaires.

Le CLI SAPI a été publié pour la première fois avec la version PHP 4.2.0, mais il était expérimental, et devait être explicitement activé avec l'option `--enable-cli` lorsque vous exécutez le script `./configure` . Depuis PHP 4.3.0, le CLI SAPI n'est plus expérimental, et l'option `--enable-cli` est activée par défaut. Vous pouvez utiliser l'option `--disable-cli` pour le désactiver.

Depuis PHP 4.3.0, le nom, l'emplacement et l'existence des binaires CLI/CGI vont dépendre de la façon dont PHP est installé sur votre système. Par défaut, en exécutant `make` , les deux binaires CGI et CLI sont compilés et nommés respectivement `sapi/cgi/php` et `sapi/cli/php` dans votre répertoire source PHP. Vous remarquerez que les deux se nomment `php`. Ce qui se passe ensuite pendant le `make install` dépend de votre ligne de configuration. Si un module SAPI, apxs par exemple, a été choisi pendant la configuration, ou que l'option `--disable-cgi` a été activée, le CLI est copié dans `{PREFIX}/bin/php` pendant le `make install` . Si, par exemple, `--with-apxs` figure dans votre ligne de configuration, le CLI est copié dans `{PREFIX}/bin/php` pendant le `make install` , sinon c'est le CGI qui y est placé. Si vous voulez forcer l'installation du binaire CGI, lancez `make install-cli` après le `make install` . Sinon, vous pouvez aussi spécifier `--disable-cgi` dans votre ligne de configuration.

Note
Du fait que les deux options <code>--enable-cli</code> et <code>--enable-cgi</code> sont activées par défaut, avoir simplement <code>--enable-cli</code> dans votre ligne de configuration n'implique pas nécessairement que le CLI soit renommé en <code>{PREFIX}/bin/php</code> pendant le <code>make install</code> .

Les paquets Windows entre PHP 4.2.0 et PHP 4.2.3 installaient le CLI en tant que `php-cli.exe` et laissaient le CGI en tant que `php-cgi.exe` dans le même répertoire. Depuis PHP 4.3.0, le paquet Windows installe le CLI en tant que `php.exe` dans un répertoire à part nommé `cli` , donc `cli/php.exe` . Depuis PHP 5, le CLI est installé dans le répertoire principal en tant que `php.exe` . La version CGI est nommée quand à elle `php-cgi.exe` .

Depuis PHP 5, un nouveau fichier php-win.exe est installé. C'est l'équivalent de la version CLI à ceci près qu'il n'affiche rien et ainsi ne fait pas apparaître de console (aucune fenêtre "dos" n'apparaît à l'écran). Ce comportement est similaire à celui de php-gtk. Vous pouvez l'activer avec l'option --enable-cli-win32 .

Note
Quel SAPI est installé?
A partir d'un interpréteur de commande, lancer php -v vous dira si php est en version CGI ou CLI. Vous pouvez aussi consulter la fonction <code>php_sapi_name</code> et la constante <code>PHP_SAPI</code> .
Note
Une page man de manuel Unix a été ajoutée avec PHP 4.3.2. Vous pouvez la consulter en tapant man php dans votre interpréteur de commande.

Les différences les plus notables entre le CLI SAPI et les SAPI sont :

- Contrairement au CGI SAPI , aucun en-tête HTTP n'est écrit dans le résultat.

Bien que le CGI SAPI fournisse un moyen de supprimer les en-têtes HTTP, il n'y a pas moyen d'activer les en-têtes HTTP dans le CLI SAPI .

CLI est lancé en mode silencieux par défaut, bien que les options -q et --no-header soient gardées pour rester compatible avec les anciennes versions CGI.

Il ne change pas le répertoire courant en celui du script. (les options -C et --no-chdir sont gardées par souci de compatibilité)

Messages d'erreurs en texte brut (pas de formatage HTML).

- Il y a plusieurs directives du php.ini qui sont ignorées par le CLI SAPI , car elles n'ont pas de sens, en environnement shell :

Directive	Valeur par défaut pour CLI SAPI	Commentaire
<code>html_errors</code>	FALSE	Il peut être bien difficile de lire les messages d'erreur sur un terminal lorsqu'ils sont noyés dans des balises HTML sans grand intérêt. Par conséquent, cette directive est forcée à FALSE .
<code>implicit_flush</code>	TRUE	Il est souhaitable que tout affichage en provenance de <code>print</code> , <code>echo</code> et consorts soit immédiatement affiché dans le terminal, et non pas placé dans un buffer quelconque. Vous pouvez toujours utiliser la bufferisation de sortie si vous voulez retarder un affichage, ou bien en manipuler le contenu une dernière fois.
<code>max_execution_time</code>	0 (sans limite)	Etant donné les possibilités infinies de PHP en environnement shell, le temps d'exécution maximal d'un script PHP a été rendu illimité. Alors que les scripts destinés au web doivent s'accomplir en une fraction de seconde, il arrive que les scripts shell requièrent bien plus de temps.
<code>register_argc_argv</code>	TRUE	En donnant la valeur de TRUE à cette directive, vous aurez toujours accès à la variable <code>argc</code> (représentant le nombre d'arguments passés à l'application) et <code>argv</code> (le tableau

	contenant les arguments passés) dans le CLI SAPI . Depuis PHP 4.3.0, les variables \$argc et \$argv sont définies et remplies avec les valeurs appropriées, en utilisant CLI SAPI . Avant cette version, la création de ces variables était liée au comportement des versions CGI et MODULE , qui requièrent l'activation de la directive <u>register_globals</u> . Indépendamment de la version ou de la valeur de register_globals, vous pouvez toujours accéder à <u>\$ _SERVER</u> et \$HTTP_SERVER_VARS . Par exemple : <u>\$ _SERVER['argv']</u>
--	--

Note
Ces directives ne peuvent pas être initialisées avec d'autres valeurs dans le fichier php.ini ou par une autre méthode. C'est une limitation, car ces valeurs par défaut s'appliquent une fois que tous les autres fichiers de configuration ont été analysés. Cependant, ces valeurs peuvent être modifiées durant l'exécution (ce qui n'est pas logique pour certaines directives, comme register_argc_argv).

- Pour faciliter le travail en environnement shell, les constantes suivantes sont définies :

Constante	Description
STDIN	Un descripteur de fichier déjà disponible vers stdin . Cela évite de l'ouvrir avec <pre><?php \$stdin = fopen('php://stdin', 'r'); ?></pre> <p>Si vous voulez lire une seule ligne depuis stdin , vous pouvez utiliser :</p> <pre><?php \$line = trim(fgets(STDIN)); // lit une seule ligne depuis STDIN fscanf(STDIN, "%d\n", \$number); // lit les nombres depuis STDIN ?></pre>
STDOUT	Un descripteur de fichier déjà disponible vers stdout . Cela évite de l'ouvrir avec <pre><?php \$stdout = fopen('php://stdout', 'w'); ?></pre>
STDERR	Un descripteur de fichier déjà disponible vers stderr . Cela évite de l'ouvrir avec <pre><?php \$stderr = fopen('php://stderr', 'w'); ?></pre>

Etant donné ce qui précède, vous n'avez pas besoin d'ouvrir un flux vers stderr par vous-même, mais vous pouvez utiliser cette constante directement, comme un descripteur de fichier :

```
php -r 'fwrite(STDERR, "stderr\n");'
```

Vous n'avez pas non plus à fermer explicitement ces fichiers, PHP s'en chargera automatiquement à la fin du script.

- Le CLI SAPI **ne transforme pas** le dossier courant en dossier d'exécution du script!

Exemple de différence entre CGI SAPI et CLI SAPI :

```
<?php
// Un test simple : affiche le dossier d'exécution */
echo getcwd(), "\n";
?>
```

Lorsque vous utilisez la version CGI , l'affichage sera :

```
$ pwd
/tmp
```

```
$ php-cgi -f autre_dossier/test.php
/tmp/autre_dossier
```

Cela montre clairement que PHP modifie le dossier courant, et utilise le dossier du script exécuté.

En utilisant le CLI SAPI , on obtient :

```
$ pwd
/tmp
```

```
$ php -f autre_dossier/test.php
/tmp
```

Cela donne beaucoup plus de souplesse lorsque vous rédigez des scripts shell avec PHP.

Note
Le CGI SAPI se comporte de la même façon que le CLI SAPI , en lui passant l'option -C , lorsque vous l'invoquez en ligne de commande.

La liste des options de ligne de commande fournies par PHP est disponible à n'importe quel moment en exécutant PHP avec l'option -h :

```
Usage: php [options] [-f] <file> [--] [args...]
php [options] -r <code> [--] [args...]
php [options] [-B <begin_code>] -R <code> [-E <end_code>] [--] [args...]
php [options] [-B <begin_code>] -F <file> [-E <end_code>] [--] [args...]
php [options] -- [args...]
php [options] -a
```

```
-a Run interactively
-c <path>|<file> Look for php.ini file in this directory
-n No php.ini file will be used
-d foo[=bar] Define INI entry foo with value 'bar'
-e Generate extended information for debugger/profiler
-f <file> Parse <file>.
-h This help
-i PHP information
-l Syntax check only (lint)
-m Show compiled in modules
-r <code> Run PHP <code> without using script tags <?..?>
-B <begin_code> Run PHP <begin_code> before processing input lines
-R <code> Run PHP <code> for every input line
```

- F <file> Parse and execute <file> for every input line
- E <end_code> Run PHP <end_code> after processing all input lines
- H Hide any passed arguments from external tools.
- s Display colour syntax highlighted source.
- v Version number
- w Display source with stripped comments and whitespace.
- z <file> Load Zend extension <file>.

args... Arguments passed to script. Use -- args when first argument starts with - or script is read from stdin

Le CLI SAPI dispose de trois moyens pour lire le code du script PHP que vous voulez exécuter :

1. Indiquer à PHP d'exécuter un fichier donné :

```
php mon_script.php
```

```
php -f mon_script.php
```

Les deux méthodes (en utilisant -f ou pas) exécute le script contenu dans le fichier mon_script.php . Vous pouvez choisir n'importe quel fichier, et ces fichiers ne sont pas tenus d'utiliser l'extension .php . N'importe quelle extension peut faire l'affaire.

2. Donner du code PHP à exécuter directement en ligne de commande.

```
php -r 'print_r(get_defined_constants());'
```

Une attention particulière doit alors être apportée aux variables d'environnement, qui seront remplacées, et aux guillemets, qui ont des significations spéciales en ligne de commande.

Note
Lisez l'exemple attentivement, il n'y a ni balise d'ouverture, ni balise de fermeture! L'option -r rend caduque l'utilisation de celles-ci, et les ajouter conduirait alors à une erreur d'analyse syntaxique.

3. Alimenter l'entrée standard en code PHP (stdin).

Cela donne la possibilité de créer dynamiquement du code PHP, puis de le fournir à PHP, et enfin, de le traiter à nouveau en shell. Voici un exemple fictif :

```
$ some_application | some_filter | php | sort -u >final_output.txt
```

Il n'est pas possible de combiner ces trois modes d'exécution.

Comme toute application shell, l'exécutable PHP accepte des arguments, et votre script PHP peut aussi les recevoir. Le nombre d'arguments n'est pas limité par PHP (le shell a une limite en terme de nombre de caractères qui peuvent être passés. Généralement, vous n'atteindrez pas cette limite). Les arguments passés au script seront transmis via la variable tableau \$argv . L'index zéro contiendra toujours le nom du script appelé (qui sera - dans le cas où le code PHP arrive de l'entrée standard ou depuis la ligne de commande, passé -r). L'autre variable globale fournie est \$argc qui contient le nombre d'éléments dans le tableau \$argv (ce nombre est **différent** du nombre d'arguments passés au script).

Tant que les arguments que vous passez à votre script ne commencent pas par le caractère - , il n'y a rien de spécial à surveiller. Si vous passez des arguments à votre script qui commencent par - , cela posera des problèmes car PHP va penser qu'il doit les interpréter. Pour éviter cela, utilisez le séparateur -- . Après cet argument, tous les arguments suivants seront passés à votre script sans

être modifié ou analysé par PHP.

```
# Cela ne va pas exécuter le code, mais afficher l'aide de PHP
$ php -r 'var_dump($argv);' -h
Usage: php [options] [-f] <file> [args...]
[...]
```

```
# Cela va passer l'argument '-h' à votre script, et éviter que PHP ne le traite
$ php -r 'var_dump($argv);' -- -h
array(2) {
[0]=>
string(1) "-"
[1]=>
string(2) "-h"
}
```

Cependant, il y a une autre méthode pour utiliser PHP en script shell. Vous pouvez aussi utiliser la ligne `#!/usr/bin/php` en tout début de votre script, suivi de code PHP compris entre balise ouvrantes/fermantes. Après avoir mis les droits d'exécution sur votre script (`chmod +x test`), il peut être exécuté comme un script shell ou perl habituel :

```
#!/usr/bin/php
<?php
var_dump($argv);
?>
```

En supposant que ce fichier s'appelle `test` , dans le dossier courant, nous pouvons alors faire ceci :

```
$ chmod +x test
$ ./test -h -- foo
array(4) {
[0]=>
string(6) "./test"
[1]=>
string(2) "-h"
[2]=>
string(2) "--"
[3]=>
string(3) "foo"
}
```

Comme vous le voyez, aucune précaution n'est nécessaire pour passer des paramètres qui commencent par `-` à votre script.

Les options longues sont disponibles depuis PHP 4.3.3.

Option	Option longue	Description
-a	--interactive	Lance PHP de façon interactive. Si vous compilez PHP avec l'extension <u>Readline</u> (qui n'est pas disponible sous Windows), vous aurez un shell joli, incluant la fonctionnalité de complétion (e.g. vous pouvez commencer à taper un nom de variable, taper la touche TABULATION et PHP complètera son nom) et un historique de ce que vous entrez au clavier qui peut être consulté en utilisant les touches fléchées. Cet historique est sauvegardé dans le fichier <code>~/.php_history</code> .

		<p>Note</p> <p>Les fichiers incluent dans auto_prepend_file et auto_append_file sont analysés dans ce mode avec quelques restrictions - par exemple, les fonctions doivent être définies avant d'être appelées.</p>
-c	--php-ini	<p>Avec cette option, vous pouvez spécifier le nom du dossier dans lequel se trouve le fichier php.ini , ou encore vous pouvez spécifier un fichier de configuration (INI) directement (qui ne s'appelle pas obligatoirement php.ini) :</p> <pre>\$ php -c /custom/directory/ mon_script.php</pre> <pre>\$ php -c /custom/directory/custom-file.ini mon_script.php</pre> <p>Si vous ne spécifiez pas cette option, le fichier est recherché dans les endroits par défaut .</p>
-n	--no-php-ini	Ignore complètement php.ini. Cette option est disponible depuis PHP 4.3.0.
-d	--define	<p>Cette option permet de modifier n'importe quelle directive de configuration du fichier php.ini . La syntaxe est :</p> <pre>-d configuration_directive[=value]</pre> <p>Exemples :</p> <pre># L'omission de la valeur conduit à donner la valeur de "1" \$ php -d max_execution_time -r '\$foo = ini_get("max_execution_time"); var_dump(\$foo);' string(1) "1"</pre> <pre># Passer une valeur vide conduit à donner la valeur de "" php -d max_execution_time= -r '\$foo = ini_get("max_execution_time"); var_dump(\$foo);' string(0) ""</pre> <pre># La directive de configuration sera n'importe quelle valeur passée après le caractère '=' \$ php -d max_execution_time=20 -r '\$foo = ini_get("max_execution_time"); var_dump(\$foo);' string(2) "20" \$ php -d max_execution_time=doesntmakesense -r '\$foo = ini_get("max_execution_time"); var_dump(\$foo);' string(15) "doesntmakesense"</pre>
-e	--profile-info	Génère des informations étendues pour le profilage et le débogage.
-f	--file	Analyse et exécute le fichier donné après l'option -f . Cette option est facultative, et peut être omise. Le seul nom du fichier est suffisant.
-h et -?	--help	Avec cette option, vous pouvez obtenir des informations sur la liste courante des options de la ligne de commande, ainsi que leur description.
-i	--info	Cette option appelle la fonction phpinfo , et affiche le résultat. Si PHP ne fonctionne pas correctement, il est recommandé d'utiliser la commande php -i et de voir s'il n'y a pas d'erreurs affichées avant ou après la table d'information. N'oubliez pas que le résultat de cette option, si vous utilisez le mode CGI, est au format HTML , et donc de taille conséquente.
-l	--syntax-check	Cette option permet de faire une vérification syntaxique sur le code PHP fourni. En cas de réussite, le message No syntax errors

		<p>detected in <filename> (Aucune d'erreur de syntaxe n'ont été détectées dans <nom_de_fichier> est affiché sur la sortie standard, et le script shell retourne 0 . En cas d'erreur, le message Errors parsing <filename> (Erreurs d'analyse dans le fichier <filename>) est affiché, en plus des messages d'erreurs détectés par l'analyseur lui-même. Le script shell retourne le code 255 .</p> <p>Cette option ne détecte pas les erreurs fatales (par exemple les fonctions non définies). Utilisez -f si vous voulez tester aussi ces erreurs.</p>		
		<table border="1"> <tr> <td>Note</td> </tr> <tr> <td>Cette option ne fonctionne pas avec l'option -r .</td> </tr> </table>	Note	Cette option ne fonctionne pas avec l'option -r .
Note				
Cette option ne fonctionne pas avec l'option -r .				
-m	--modules	<p>Cette option liste les extensions PHP et Zend compilées (et chargées) :</p> <pre>\$ php -m [PHP Modules] xml tokenizer standard session posix pcre overload mysql mbstring ctype [Zend Modules]</pre>		
-r	--run	<p>Cette option permet l'exécution de PHP directement dans la ligne de commande. Les balises de PHP (<?php et ?>) ne sont pas nécessaires, et causeront une erreur d'analyse si elles sont présentes.</p>		
		<table border="1"> <tr> <td>Note</td> </tr> <tr> <td> <p>Une attention toute particulière doit être portée lors de l'utilisation de cette option de PHP, pour qu'il n'y ait pas de collision avec les substitutions de variables en ligne de commande, réalisées par le shell.</p> <p>Exemple conduisant à une erreur d'analyse :</p> <pre>\$ php -r "\$foo = get_defined_constants();" Command line code(1) : Parse error - parse error, unexpected '=' Le problème ici est que le shell (sh/bash) effectue une substitution de variables, même avec les guillemets doubles " . puisque la variable \$foo n'est probablement pas définie dans le shell, elle est remplacée par rien, ce qui fait que le code passé à PHP pour l'exécution est :</pre> <pre>\$ php -r " = get_defined_constants();" La solution de ce problème est d'utiliser les guillemets simples ' . Les variables de ces chaînes ne seront pas substituées par leurs valeurs par le shell.</pre> <pre>\$ php -r '\$foo = get_defined_constants(); var_dump(\$foo);'</pre> </td> </tr> </table>	Note	<p>Une attention toute particulière doit être portée lors de l'utilisation de cette option de PHP, pour qu'il n'y ait pas de collision avec les substitutions de variables en ligne de commande, réalisées par le shell.</p> <p>Exemple conduisant à une erreur d'analyse :</p> <pre>\$ php -r "\$foo = get_defined_constants();" Command line code(1) : Parse error - parse error, unexpected '=' Le problème ici est que le shell (sh/bash) effectue une substitution de variables, même avec les guillemets doubles " . puisque la variable \$foo n'est probablement pas définie dans le shell, elle est remplacée par rien, ce qui fait que le code passé à PHP pour l'exécution est :</pre> <pre>\$ php -r " = get_defined_constants();" La solution de ce problème est d'utiliser les guillemets simples ' . Les variables de ces chaînes ne seront pas substituées par leurs valeurs par le shell.</pre> <pre>\$ php -r '\$foo = get_defined_constants(); var_dump(\$foo);'</pre>
Note				
<p>Une attention toute particulière doit être portée lors de l'utilisation de cette option de PHP, pour qu'il n'y ait pas de collision avec les substitutions de variables en ligne de commande, réalisées par le shell.</p> <p>Exemple conduisant à une erreur d'analyse :</p> <pre>\$ php -r "\$foo = get_defined_constants();" Command line code(1) : Parse error - parse error, unexpected '=' Le problème ici est que le shell (sh/bash) effectue une substitution de variables, même avec les guillemets doubles " . puisque la variable \$foo n'est probablement pas définie dans le shell, elle est remplacée par rien, ce qui fait que le code passé à PHP pour l'exécution est :</pre> <pre>\$ php -r " = get_defined_constants();" La solution de ce problème est d'utiliser les guillemets simples ' . Les variables de ces chaînes ne seront pas substituées par leurs valeurs par le shell.</pre> <pre>\$ php -r '\$foo = get_defined_constants(); var_dump(\$foo);'</pre>				

		<pre>array(370) { ["E_ERROR"]=> int(1) ["E_WARNING"]=> int(2) ["E_PARSE"]=> int(4) ["E_NOTICE"]=> int(8) ["E_CORE_ERROR"]=> [...]</pre> <p>Si vous utilisez un shell différent de sh/bash, vous pouvez rencontrer d'autres problèmes. N'hésitez pas à ouvrir un rapport de bogues à http://bugs.php.net/ . Il est toujours très facile d'avoir des problèmes lorsque vous essayez d'inclure des variables shell dans le code, ou d'utiliser les anti-slash pour l'échappement. Vous aurez été prévenu.</p> <p>Note</p> <p>-r est disponible avec le CLI SAPI et pas avec le CGI SAPI.</p> <p>Note</p> <p>Cette option est utilisée pour des choses vraiment de base. Ainsi, quelques directives de configuration (par exemple <u>auto_prepend_file</u> et <u>auto_append_file</u>) sont ignorées dans ce mode.</p>
-B	--process-begin	PHP code to execute before processing stdin. Added in PHP 5.
-R	--process-code	Code PHP à exécuter pour chaque ligne en entrée. Ajouté en PHP 5. Il y a deux variables spéciales de disponible dans ce mode : \$argn et \$argi . \$argn doit contenir la ligne PHP traitée à ce moment donné, tandis que \$argi doit contenir le numéro de la ligne.
-F	--process-file	Fichier PHP à exécuter pour chaque ligne en entrée. Ajouté en PHP 5.
-E	--process-end	Code PHP à exécuter après avoir effectué l'entrée. Ajouté en PHP 5. Exemple d'utilisation des options -B , -R et -E pour compter le nombre de lignes d'un projet. \$ find my_proj php -B '\$l=0;' -R '\$l += count(@file(\$argn));' -E 'echo "Total Lines: \$l\n";' Total Lines: 37328
-s	--syntax-highlight et --syntax-highlighting	Affiche le code avec la colorisation syntaxique. Cette option utilise le mécanisme interne pour analyser le fichier, et produire une version colorisée du code source, au format HTML . Notez que cette option ne fait que générer un bloc HTML , avec les balises HTML <code> [...] </code> , sans en-têtes HTML . Note Cette option ne fonctionne pas avec l'option -r .
-v	--version	Affiche les versions de PHP, PHP SAPI, et Zend sur la sortie standard. Par exemple : \$ php -v PHP 4.3.0 (cli), Copyright (c) 1997-2002 The PHP Group

		Zend Engine v1.3.0, Copyright (c) 1998-2002 Zend Technologies		
-w	--strip	Affiche la source sans les commentaires et les espaces. <table border="1"> <tr> <td>Note</td> </tr> <tr> <td>Cette option ne fonctionne pas avec l'option -r .</td> </tr> </table>	Note	Cette option ne fonctionne pas avec l'option -r .
Note				
Cette option ne fonctionne pas avec l'option -r .				
-z	--zend-extension	Charge une extension Zend. Si et seulement si un fichier est fourni, PHP essaie de charger cette extension dans le dossier courant par défaut des bibliothèque sur votre système (généralement spécifié avec /etc/ld.so.conf sous Linux). Passer un nom de fichier avec le chemin complet fera que PHP utilisera ce fichier, sans recherche dans les dossiers classiques. Un chemin de dossier relatif indiquera à PHP qu'il doit chercher les extensions uniquement dans ce dossier.		

L'exécutable PHP peut être utilisé pour exécuter des scripts indépendants du serveur web. Si vous êtes sur un système Unix, il est recommandé d'ajouter la ligne spéciale (prononcer " shebang ") en début de script, de le rendre exécutable de manière à ce que le système sache quel programme doit exécuter le script. Sous Windows, vous pouvez associer l'exécutable php.exe avec le double-clic sur les fichiers d'extension .php , ou bien vous pouvez faire un fichier batch pour exécuter le script grâce à PHP. La première ligne utilisée dans le monde Unix ne perturbera pas l'exécution sous Windows, ce qui rend les scripts facilement portables. Un exemple complet est disponible ci-dessous :

<p>Script prévu pour être exécuté en ligne de commande (script.php)</p> <pre>#!/usr/bin/php <?php if (\$argc != 2 in_array(\$argv[1], array('--help', '-help', '-h', '-?'))) { ?> C'est une ligne de commande à une option. Utilisation : <?php echo \$argv[0]; ?> <option> <option> peut être un mot que vous souhaitez afficher. Avec les options --help, -help, -h, et -?, vous obtiendrez cette aide. <?php } else { echo \$argv[1]; } ?></pre>

Dans le script ci-dessus, nous utilisons la première ligne pour indiquer que le fichier doit être exécuté par PHP. Nous travaillons avec une version CLI, donc aucun en-tête HTTP n'est affiché. Il y a deux variables que vous pouvez utiliser avec les applications de ligne de commande : \$argc et \$argv . La première est le nombre d'arguments plus un (le nom du script qui est exécuté). La seconde est un tableau contenant les arguments, commençant avec le nom du script en élément 0 (\$argv[0]).

Dans notre exemple, nous avons vérifié qu'il y a plus ou moins d'un argument. De plus, si cet argument est --help , -help , -h ou -? , nous avons affiché un message d'aide, ainsi que le nom du script. Nous nous recevons un autre argument, cet argument est affiché dans le terminal.

Pour exécuter le script ci-dessus sous Unix, vous devez le rendre exécutable, puis l'appeler avec une commande comme : script.php echothis ou script.php -h . Sous Windows, vous pouvez faire un fichier batch pour cela :

Fichier batch pour exécuter un script PHP en ligne de commande (script.bat)

@C:\php\php.exe script.php %1 %2 %3 %4
--

Si vous avez nommé le programme ci-dessus script.php , et que vous avez votre exécutable php.exe situé à C:\php\php.exe , ce fichier batch l'exécutera avec les options que vous lui passez : script.bat echothis ou script.bat -h .

Voir aussi l'extension [Readline](#) , qui dispose de nombreuses fonctions pour améliorer la convivialité de vos applications en ligne de commande.

8 Référence des fonctions

8.1 Fonctions Apache

8.1.1 Introduction

Ces fonctions sont disponibles lorsque PHP est utilisé comme module Apache.

Note
Depuis PHP 4.3.2, la variable serveur PATH_TRANSLATED n'est plus configurée automatiquement sous Apache 2 SAPI au contraire de la situation dans Apache 1, où elle est configurée à la même valeur que la variable serveur SCRIPT_FILENAME lorsqu'elle n'est pas peuplée par Apache. Cette modification a été apportée pour être conforme aux spécifications CGI qui fait que la variable serveur PATH_TRANSLATED doit uniquement exister que si PATH_INFO est définie.
Les utilisateurs d'Apache 2 devraient utiliser <code>AcceptPathInfo = On</code> au lieu de <code>httpd.conf</code> pour définir PATH_INFO .

8.1.2 Installation

Pour l'installation de PHP sous Apache, lisez le [chapitre d'installation](#).

8.1.3 Configuration à l'exécution

Le comportement du module PHP d'Apache est configurable dans le fichier `php.ini`. Les configurations du `php.ini` peuvent être remplacées par l'option `php_flag` dans le fichier de configuration du serveur, ou dans les fichiers locaux `.htaccess`.

Désactiver l'analyse des fichiers PHP dans un dossier avec <code>.htaccess</code>
<code>php_flag engine off</code>

Nom	Par défaut	Modifiable	Historique
<code>engine</code>	"1"	PHP_INI_ALL	Disponible depuis PHP 4.0.5.
<code>child_terminate</code>	"0"	PHP_INI_ALL	Disponible depuis PHP 4.0.5.
<code>last_modified</code>	"0"	PHP_INI_ALL	Disponible depuis PHP 4.0.5.
<code>xbit_hack</code>	"0"	PHP_INI_ALL	Disponible depuis PHP 4.0.5.

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

`engine` booléen

Active ou non l'interpréteur PHP. Cette directive est utile uniquement pour le module Apache. Elle est utilisée par les sites qui souhaitent activer ou désactiver PHP, au cas par cas, par dossier ou par dossier virtuel. En utilisant `engine off` au bon endroit dans le fichier `httpd.conf`, PHP peut être activé ou désactivé.

`child_terminate` boolean

Spécifie si les scripts PHP peuvent réclamer la fin des processus fils en fin de requête. Voir aussi [apache_child_terminate](#) .

last_modified boolean

Envoie la date de modification des scripts PHP dans l'en-tête HTTP Last-Modified: .

xbithack boolean

Analyse avec PHP les fichiers exécutables, indépendamment de leur extension.

8.1.4 Types de ressources

Cette extension ne définit aucune ressource.

8.1.5 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [apache_child_terminate](#)
- [apache_get_modules](#)
- [apache_get_version](#)
- [apache_getenv](#)
- [apache_lookup_uri](#)
- [apache_note](#)
- [apache_request_headers](#)
- [apache_reset_timeout](#)
- [apache_response_headers](#)
- [apache_setenv](#)
- [ascii2ebcdic](#)
- [ebcdic2ascii](#)
- [getallheaders](#)
- [virtual](#)

8.1.7 `apache_get_modules()` : Retourne la liste des modules Apache chargés

array `apache_get_modules` (void)

Retourne la liste des modules Apache chargés.

8.1.8 `apache_get_version()` : Récupère la version d'Apache

string `apache_get_version` (void)

Récupère la version d'Apache.

8.1.9 `apache_getenv()` : Obtient une variable `subprocess_env` d'Apache

string `apache_getenv` (string `variable` , *bool* `walk_to_top`)

Récupère une variable d'environnement d'Apache comme spécifié par `variable` .

Cette fonction requiert Apache 2, autrement elle est indéfinie.

8.1.10 `apache_lookup_uri()` : Effectue une requête partielle pour l'URI spécifiée et renvoie toutes les informations

object `apache_lookup_uri` (string `filename`)

`apache_lookup_uri` effectue une requête partielle pour l'URI spécifiée. Cette requête permet de récupérer toutes les informations importantes à propos de la ressource concernée.

Cette fonction n'est supportée que si PHP est installé en tant que module d'Apache.

8.1.11 `apache_note()` : Affiche ou affecte le paramètre "apache request notes"

string `apache_note` (string `note_name` , *string* `note_value`)

`apache_note` est une fonction spécifique au serveur Apache. Cette fonction affecte ou renvoie la valeur de la variable contenue dans la table notes d'Apache.

8.1.12 `apache_request_headers()` : Récupère tous les en-têtes HTTP de la requête

array `apache_request_headers` (void)

Récupère tous les en-têtes HTTP de la requête.

Cette fonction n'est supportée que si PHP est installé en tant que module d'Apache.

8.1.13 `apache_reset_timeout()` : Remet à sa position initiale le temporisateur d'Apache

bool `apache_reset_timeout` (void)

`apache_reset_timeout` remet à sa position initiale le temporisateur d'Apache, qui vaut par défaut 300 secondes. Avec des appels à `set_time_limit(0)`; `ignore_user_abort(true)` et des appels périodiques à `apache_reset_timeout` , Apache peut, théoriquement, fonctionner sans arrêt.

Cette fonction nécessite Apache 1.

8.1.14 `apache_response_headers()` : Récupère tous les en-têtes de réponse HTTP

array `apache_response_headers` (void)

Récupère tous les en-têtes de réponse HTTP.

8.1.15 `apache_setenv()` : Modifie une variable de `subprocess_env` Apache

bool `apache_setenv` (string `variable` , string `value` , bool `walk_to_top`)

`apache_setenv` définit la valeur de la variable d'environnement Apache spécifiée par le paramètre `variable` .

Note
Lors de la définition d'une variable d'environnement Apache, la variable <code>\$_SERVER</code> correspondante n'est pas modifiée.

8.1.16 `ascii2ebcdic()` : Transforme une chaîne ASCII en EBCDIC

int `ascii2ebcdic` (string `ascii_str`)

`ascii2ebcdic` est une fonction spécifique à Apache, qui n'est disponible que sur les systèmes d'exploitation qui gèrent le format EBCDIC (OS/390, BS2000). Elle traduit la chaîne ASCII `ascii_str` en son équivalent EBCDIC (avec protection des données binaires) et retourne le résultat.

8.1.17 `ebcdic2ascii()` : Transforme une chaîne EBCDIC en ASCII

int `ebcdic2ascii` (string `ebcdic_str`)

`ebcdic2ascii` est une fonction spécifique à Apache, qui n'est disponible que sur les OS qui gèrent le format EBCDIC (OS/390, BS2000). Elle traduit la chaîne EBCDIC `ebcdic_str` en son équivalent ASCII (avec protection des données binaires) et retourne le résultat.

8.1.18 `getallheaders()` : Récupère tous les en-têtes de la requête HTTP

array `getallheaders` (void)

Récupère tous les en-têtes de la requête HTTP.

Cette fonction est un alias pour `apache_request_headers` . Lisez la documentation de `apache_request_headers` pour plus d'informations sur le fonctionnement de cette fonction.

Cette fonction n'est supportée que si PHP est installé en tant que module d'Apache.

8.1.19 **virtual()** : Effectue une sous-requête Apache

bool **virtual** (string filename)

virtual est une fonction spécifique au serveur Apache. Elle est similaire à la directive " <!--#include virtual...--> " lorsque vous utilisez le mod_include d'Apache. Cette fonction effectue une sous-requête Apache. C'est très utile lorsque vous voulez analyser des scripts CGI, des fichiers .shtml ou n'importe quel autre type de fichier à travers le serveur Apache. Il est à noter que lors de l'utilisation avec des scripts CGI, ces derniers doivent générer un en-tête valide, c'est-à-dire, au minimum un en-tête Content-Type .

Pour exécuter une sous-requête, tous les tampons sont arrêtés et vidés au navigateur, les en-têtes restants le sont aussi.

8.2 Débogueur avancé pour PHP

8.2.1 Introduction

APD est un débogueur avancé pour PHP (Advanced PHP Debugger). Il a été écrit pour fournir des capacités de profilage et de débogage du code PHP tout comme la possibilité d'afficher un retour de trace de la pile. APD supporte le débogage interactif, mais, par défaut, il écrit les données dans des fichiers de trace. APD fournit également la notation basée par événement, de sorte que les niveaux d'information (incluant les appels de fonctions, le passage d'arguments, le délai d'exécution, etc.) peuvent être activés ou non pour chaque script.

Attention

APD est une extension Zend, qui modifie la façon interne de gestion des appels de fonctions PHP et peut ou ne pas être compatible avec d'autres extensions Zend (par exemple avec le Zend Optimizer).

8.2.2 Installation

APD est actuellement disponible en tant qu'extension PECL depuis <http://pecl.php.net/package/apd>. Assurez-vous d'avoir installé la version CGI de PHP et qu'elle soit disponible dans votre PATH courant tout au long de votre script PHP.

Exécutez la commande suivante pour télécharger, construire et installer la dernière version stable d'APD :

```
pear install apd
```

Ceci installera automatiquement le module APD Zend dans votre dossier d'extensions de PHP. Il n'est pas obligatoire de le conserver à cet endroit ; vous pouvez stocker le module dans n'importe quel dossier que PHP peut lire, à partir du moment où vous paramétrez correctement le paramètre `zend_extension`.

Les utilisateurs de Windows peuvent télécharger la bibliothèque dll `php_apd.dll` depuis http://snaps.php.net/win32/PECL_STABLE/.

Dans votre fichier INI, ajoutez les lignes suivantes :

```
zend_extension = /chemin/absolu/vers/apd.so
apd.dumpdir = /chemin/absolu/vers/dossier/trace
apd.statement_tracing = 0
```

En fonction de votre installation de PHP, le dossier `zend_extension` peut être l'un de la liste suivante :

```
zend_extension          (non ZTS, non debug build)
zend_extension_ts      (   ZTS, non debug build)
zend_extension_debug   (non ZTS,   debug build)
zend_extension_debug_ts (   ZTS,   debug build)
```

8.2.3 Compilation sous un environnement Win32

Pour compiler APD sous Windows, vous avez besoin d'un environnement de compilation PHP comme décrit sur <http://php.net/> -- simplement, cela nécessite que vous possédiez Microsoft Visual

C++, win32build.zip, bison/flex certains savent comment les récupérer et les faire fonctionner. Assurez-vous également que `apd.dsp` a bien des fins de lignes façon DOS ; si les fins de lignes sont du style Unix, Microsoft Visual C++ vous le fera remarquer.

8.2.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable	Historique
<code>apd.dumpdir</code>	NULL	PHP_INI_ALL	
<code>apd.statement-tracing</code>	"0"	PHP_INI_ALL	

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

`apd.dumpdir` chaîne de caractères

Spécifie le dossier dans lequel APD écrit les fichiers de profilage. Vous pouvez spécifier un chemin absolu ou relatif.

Vous pouvez spécifier un dossier différent comme argument de `apd_set_pprof_trace`.

`apd.statement_tracing` boolean

Active ou désactive les traces à la ligne. En activant cette option (valeur de 1), l'application sera considérablement ralentie.

8.2.5 Types de ressources

Cette extension ne définit aucune ressource.

8.2.6 Constantes pré-définies

Cette extension ne définit aucune constante.

8.2.7 Comment utiliser APD dans vos scripts ?

A la première ligne de vos scripts PHP, appelez la fonction `apd_pprof_trace` pour commencer la trace :

```
apd_set_pprof_trace();
```

Vous pouvez ajouter cette ligne n'importe où dans votre script, mais si vous ne commencez pas à tracer au début de votre script, vous désactivez les données de profil qui pourraient autrement vous mener à un goulot d'étranglement d'exécution.

Maintenant, exécutez votre script. Le contenu de la sortie devrait être écrit dans le fichier `apd.dumpdir/pprof_pid.ext`. Array

Pour afficher les données de profile formatées, exécutez la commande `pprof` avec les options de trie et d'affichage de votre choix. L'affichage formatée pourra ressembler à cela :

```
bash-2.05b$ pprof -R /tmp/pprof.22141.0
```

```
Trace for /home/dan/testapd.php
Total Elapsed Time = 0.00
Total System Time = 0.00
Total User Time = 0.00
```

```
Real User System secs/ cumm
%Time (excl/cumm) (excl/cumm) (excl/cumm) Calls call s/call Memory Usage Name
-----
100.0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1 0.0000 0.0009 0 main
56.9 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1 0.0005 0.0005 0 apd_set_pprof_trace
28.0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 10 0.0000 0.0000 0 preg_replace
14.3 0.00 0.00 0.00 0.00 0.00 0.00 0.00 10 0.0000 0.0000 0 str_replace
```

L'option -R utilisée dans cet exemple trie la table de profil par la durée réelle qu'une fonction donnée coûte au script. La colonne "cumm call" révèle le nombre d'appel de chaque fonction et la colonne "s/call", le nombre de secondes chaque appel à la fonction nécessite, en moyenne.

Pour générer un fichier d'appel sous forme d'arbre que vous pourrez importer dans l'application KCacheGrind (application d'analyse de profile), exécutez la commande `pprof2calltree` .

8.2.8 Informations de contact

Si vous avez des commentaires, des corrections de bogues ou si vous voulez développer des améliorations pour cette extension, vous pouvez envoyer un email à apd@mail.communityconnect.com . Toute aide est vraiment la bienvenue.

- [Introduction](#)
- [Installation](#)
- [Compilation sous un environnement Win32](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Comment utiliser APD dans vos scripts ?](#)
- [Informations de contact](#)
- [apd_breakpoint](#)
- [apd_callstack](#)
- [apd_clunk](#)
- [apd_continue](#)
- [apd_croak](#)
- [apd_dump_function_table](#)
- [apd_dump_persistent_resources](#)
- [apd_dump_regular_resources](#)
- [apd_echo](#)
- [apd_get_active_symbols](#)
- [apd_set_pprof_trace](#)
- [apd_set_session_trace](#)
- [apd_set_session](#)
- [apd_set_socket_session_trace](#)
- [override_function](#)
- [rename_function](#)

8.2.10 `apd_callstack()` : Retourne la pile d'appel courante dans un tableau

array `apd_callstack` (void)

`apd_callstack` retourne la pile d'appel courante dans un tableau d'éléments.

Exemple avec `apd_callstack`

```
<?php
print_r(apd_callstack());
?>
```

8.2.11 `apd_clunk()` : Lance une alerte et un callstack

void `apd_clunk` (string *warning* , string *delimiter*)

`apd_clunk` fonctionne de la même façon que son homologue Perl `Carp::cluck` . `apd_clunk` lance une alerte et un callstack en utilisant le paramètre *delimiter* comme délimiteur. La valeur par défaut de ce délimiteur est "
\n " .

Exemple avec `apd_clunk`

```
<?php
apd_clunk("Quelques alertes","<P>");
?>
```

8.2.12 `apd_continue()` : Redémarre l'interpréteur

bool `apd_continue` (int *debug_level*)

`apd_continue` envoie une instruction au socket pour redémarrer l'interpréteur.

Exemple avec `apd_continue`

```
<?php
apd_continue(0);
?>
```

8.2.13 `apd_croak()` : Lance une erreur, un callstack et sort

void `apd_croak` (string *warning* , string *delimiter*)

`apd_croak` fonctionne comme son homologue Perl `Carp::croak` . `apd_croak` lance une erreur, un callstack et sort. Le paramètre *delimiter* vaut par défaut "
\n " .

Exemple avec `apd_croak`

```
<?php
apd_croak("Quelques alertes","<P>");
?>
```

8.2.14 `apd_dump_function_table()` : Affiche la table courant de fonction

`void apd_dump_function_table (void)`

`apd_dump_function_table` affiche la table courante de fonction.

Exemple avec `apd_dump_function_table`

```
<?php
apd_dump_function_table();
?>
```

8.2.15 `apd_dump_persistent_resources()` : Retourne toutes les ressources persistantes dans un tableau

`array apd_dump_persistent_resources (void)`

`apd_dump_persistent_resources` retourne toutes les ressources persistantes dans un tableau.

Exemple avec `apd_dump_persistent_resources`

```
<?php
print_r(apd_dump_persistent_resources());
?>
```

8.2.16 `apd_dump_regular_resources()` : Retourne toutes les ressources régulières courantes dans un tableau

`array apd_dump_regular_resources (void)`

`apd_dump_regular_resources` retourne toutes les ressources régulières courantes dans un tableau.

Exemple avec `apd_dump_regular_resources`

```
<?php
print_r(apd_dump_regular_resources());
?>
```

8.2.17 `apd_echo()` : Ecrit dans le socket de débogage

`bool apd_echo (string output)`

`apd_echo` envoie via le socket une demande d'information concernant le script exécuté.

Exemple avec `apd_echo`

```
<?php
apd_echo($i);
?>
```

8.2.18 `apd_get_active_symbols()` : Récupère un tableau contenant les noms de variables courantes de portées locales

array **apd_get_active_symbols** ()

apd_get_active_symbols récupère les noms de toutes les variables définies dans le contexte actif (et non leurs valeurs).

Exemple avec **apd_get_active_symbols**

```
<?php
apd_echo(apd_get_active_symbols());
?>
```

8.2.19 apd_set_pprof_trace() : Démarre la session de débogage APD

void **apd_set_pprof_trace** (*string* *dump_directory*)

apd_set_pprof_trace démarre la session de débogage dans le dossier *dump_directory* /pprof_{process_id}. Si *dump_directory* n'est pas fourni, la directive `apd.dumpdir` du `php.ini` sera utilisée.

Exemple avec **apd_set_pprof_trace**

```
<?php
apd_set_pprof_trace();
?>
```

8.2.20 apd_set_session_trace() : Démarre la session de débogage

void **apd_set_session_trace** (int *debug_level* , *string* *dump_directory*)

apd_set_session_trace démarre le débogage de { *dump_directory* }/apd_dump_{process_id}. Si le paramètre *dump_directory* n'est pas défini, la valeur du paramètre de configuration `apd.dumpdir` du `php.ini` sera utilisée.

Le paramètre *debug_level* est un entier qui est formé en ajoutant les valeurs suivantes :

FUNCTION_TRACE	1	
ARGS_TRACE	2	
ASSIGNMENT_TRACE	4	
STATEMENT_TRACE	8	
MEMORY_TRACE	16	
TIMING_TRACE	32	
SUMMARY_TRACE	64	

Je vous recommande de ne jamais utiliser `MEMORY_TRACE` . Il est vraiment très lent et ne semble pas être très précis. `ASSIGNMENT_TRACE` n'est pas actuellement implémentée. Donc, pour activer toutes les fonctionnalités de trace, (`TIMING` , `FUNCTIONS` , `ARGS` `SUMMARY` (comme `trace -c`), utilisez la valeur 99.

Exemple avec **apd_set_session_trace**

```
<?php
apd_set_session_trace(99);
?>
```

8.2.21 `apd_set_session()` : Modifie ou définit le degré de débogage courant

void **apd_set_session** (int debug_level)

`apd_set_session` peut être utilisé pour augmenter ou diminuer le degré de débogage dans un endroit particulier de votre application. Le paramètre `debug_level` est un entier qui est formé en ajoutant les valeurs suivantes :

FUNCTION_TRACE	1
ARGS_TRACE	2
ASSIGNMENT_TRACE	4
STATEMENT_TRACE	8
MEMORY_TRACE	16
TIMING_TRACE	32
SUMMARY_TRACE	64

Exemple avec `apd_set_session`

```
<?php
apd_set_session(9);
?>
```

8.2.22 `apd_set_socket_session_trace()` : Démarre la session de débogage à distance

bool **apd_set_socket_session_trace** (string ip_address_or_unix_socket_file , int socket_type , int port , int debug_level)

`apd_set_socket_session_trace` se connecte au serveur tcp (e.g. `tcplisten`) spécifié par l'IP ou une socket Unix (comme un fichier) et envoie des données de débogage au socket. Vous pouvez utiliser n'importe quel port en le spécifiant dans le paramètre `port` mais les nombres les plus élevés sont meilleurs que les nombres plus petits qui peuvent être utilisés par d'autres services du système.

Le paramètre `socket_type` peut être `APD_AF_UNIX` (pour les sockets basés sur des fichiers) ou `APD_AF_INET` (pour le standard tcp/ip).

Le paramètre `debug_level` est un entier qui est formé en ajoutant les valeurs suivantes :

FUNCTION_TRACE	1
ARGS_TRACE	2
ASSIGNMENT_TRACE	4
STATEMENT_TRACE	8
MEMORY_TRACE	16
TIMING_TRACE	32
SUMMARY_TRACE	64

Je ne vous recommande absolument pas de définir la valeur à 0 pour commencer ainsi que d'utiliser les méthodes de points de contrôle pour démarrer le débogage à un point spécifique du fichier.

Exemple avec `apd_set_socket_session_trace`

```
<?php
apd_set_socket_session_trace("127.0.0.1",APD_AF_INET,7112,0);
?>
```

8.2.23 `override_function()` : Dépasse les fonctions intégrées

`bool override_function (string function_name , string function_args , string function_code)`

`override_function` a une syntaxe similaire à la fonction `create_function` . `override_function` dépasse les fonctions intégrées (les remplace dans la table des symboles).

Exemple avec `override_function`

```
<?php
override_function('test', '$a,$b', 'echo "ON FAIT UN TEST"; return $a * $b;');
?>
```

8.2.24 `rename_function()` : Renomme une fonction intégrée dans la table des fonctions globales

`bool rename_function (string original_name , string new_name)`

`rename_function` renomme une fonction intégrée dans la table des fonctions globale. Utile pour dépasser une fonction intégrée temporairement.

Exemple avec `rename_function`

```
<?php
rename_function('mysql_connect', 'debug_mysql_connect' );
?>
```

8.3 Tableaux

8.3.1 Introduction

Ces fonctions vous permettent de manipuler et de traiter les tableaux de nombreuses façons. Les tableaux sont très efficaces dès qu'il s'agit de stocker, gérer et traiter des données en groupe.

Les tableaux simples et multi-dimensionnels sont supportés et peuvent être créés par l'utilisateur, ou par une fonction. Il y a des fonctions spécifiques qui remplissent des tableaux à partir de résultats de requêtes, et de nombreuses fonctions retournent un tableau.

Reportez-vous à la section [tableaux](#) du manuel pour des explications détaillées sur l'implémentation des tableaux en PHP. Voyez aussi [les opérateurs de tableaux](#) pour d'autres moyens de manipulations.

8.3.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.3.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.3.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.3.5 Types de ressources

Cette extension ne définit aucune ressource.

8.3.6 Constantes pré-définies

Les constantes listées ici sont toujours disponibles dans PHP.

CASE_LOWER (integer)

CASE_LOWER est utilisé avec [array_change_key_case](#) et sert à convertir tous les index d'un tableau en minuscules. C'est aussi le comportement par défaut de [array_change_key_case](#) .

CASE_UPPER (integer)

CASE_UPPER est utilisé avec [array_change_key_case](#) et sert à convertir tous les index d'un tableau en majuscule.

Constantes d'ordre de tri

SORT_ASC (integer)

`SORT_ASC` trie en ordre ascendant
`SORT_DESC` (integer)
`SORT_DESC` trie en ordre descendant

Autres constantes d'ordre de tri

`SORT_REGULAR` (integer)
`SORT_REGULAR` compare normalement les valeurs d'un tri.
`SORT_NUMERIC` (integer)
`SORT_NUMERIC` compare numériquement les valeurs d'un tri.
`SORT_STRING` (integer)
`SORT_STRING` compare alphabétiquement les valeurs d'un tri.
`SORT_LOCALE_STRING` (integer)
`SORT_LOCALE_STRING` compare alphabétiquement les valeurs d'un tri, en utilisant la configuration locale. Ajouté en PHP 5.0.2 et PHP 4.4.0.

`COUNT_NORMAL` (integer)
`COUNT_RECURSIVE` (integer)
`EXTR_OVERWRITE` (integer)
`EXTR_SKIP` (integer)
`EXTR_PREFIX_SAME` (integer)
`EXTR_PREFIX_ALL` (integer)
`EXTR_PREFIX_INVALID` (integer)
`EXTR_PREFIX_IF_EXISTS` (integer)
`EXTR_IF_EXISTS` (integer)
`EXTR_REFS` (integer)

8.3.7 Voir aussi

Voir aussi [is_array](#) , [explode](#) , [implode](#) , [split](#) , [preg_split](#) et [unset](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Voir aussi](#)
- [array_change_key_case](#)
- [array_chunk](#)
- [array_combine](#)
- [array_count_values](#)
- [array_diff_assoc](#)
- [array_diff_key](#)
- [array_diff_uassoc](#)
- [array_diff_ukey](#)
- [array_diff](#)
- [array_fill](#)
- [array_filter](#)
- [array_flip](#)
- [array_intersect_assoc](#)
- [array_intersect_key](#)
- [array_intersect_uassoc](#)

- [array_intersect_ukey](#)
- [array_intersect](#)
- [array_key_exists](#)
- [array_keys](#)
- [array_map](#)
- [array_merge_recursive](#)
- [array_merge](#)
- [array_multisort](#)
- [array_pad](#)
- [array_pop](#)
- [array_product](#)
- [array_push](#)
- [array_rand](#)
- [array_reduce](#)
- [array_reverse](#)
- [array_search](#)
- [array_shift](#)
- [array_slice](#)
- [array_splice](#)
- [array_sum](#)
- [array_udiff_assoc](#)
- [array_udiff_uassoc](#)
- [array_udiff](#)
- [array_uintersect_assoc](#)
- [array_uintersect_uassoc](#)
- [array_uintersect](#)
- [array_unique](#)
- [array_unshift](#)
- [array_values](#)
- [array_walk_recursive](#)
- [array_walk](#)
- [array](#)
- [arsort](#)
- [asort](#)
- [compact](#)
- [count](#)
- [current](#)
- [each](#)
- [end](#)
- [extract](#)
- [in_array](#)
- [key](#)
- [krsort](#)
- [ksort](#)
- [list](#)
- [natcasesort](#)
- [natsort](#)
- [next](#)
- [pos](#)
- [prev](#)
- [range](#)
- [reset](#)
- [rsort](#)
- [shuffle](#)

- [sizeof](#)
- [sort](#)
- [uasort](#)
- [uksort](#)
- [usort](#)

8.3.9 `array_chunk()` : Sépare un tableau en tableaux de taille inférieure

`array array_chunk (array input , int size , bool preserve_keys)`

`array_chunk` sépare le tableau `input` en plusieurs tableaux de taille `size` . Il est aussi possible que le dernier tableau ait moins de valeurs. Le résultat est un tableau multidimensionnel, indexé numériquement.

En passant la valeur `TRUE` au paramètre optionnel `preserve_keys` , vous pouvez forcer PHP à préserver les clés originales du tableau `input` . Si vous utilisez la valeur par défaut (`FALSE`), de nouveaux index numériques seront fournis, commençant à 0.

Exemple avec `array_chunk`

```
<?php
$input_array = array('a', 'b', 'c', 'd', 'e');
print_r(array_chunk($input_array, 2));
print_r(array_chunk($input_array, 2, TRUE));
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
  [0] => Array
  (
    [0] => a
    [1] => b
  )
  [1] => Array
  (
    [0] => c
    [1] => d
  )
  [2] => Array
  (
    [0] => e
  )
)
Array
(
  [0] => Array
  (
    [0] => a
```

```
[1] => b
)
[1] => Array
(
[2] => c
[3] => d
)
[2] => Array
(
[4] => e
)
)
```

8.3.10 `array_combine()` : Crée un tableau à partir de deux autres tableaux

`array` **`array_combine`** (`array` keys , `array` values)

`array_combine` retourne un `array` , dont les clés sont les valeurs de `keys` , et les valeurs sont les valeurs de `values` .

`array_combine` retourne `FALSE` si le nombre d'éléments de chaque tableau n'est pas le même, ou si les tableaux sont vides.

Exemple avec `array_combine`

```
<?php
$a = array('vert', 'rouge', 'jaune');
$b = array('avocat', 'pomme', 'banane');
$c = array_combine($a, $b);

print_r($c);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[vert] => avocat
[rouge] => pomme
[jaune] => banane
)
```

Voir aussi [array_merge](#) , [array_walk](#) et [array_values](#) .

8.3.11 `array_count_values()` : Compte le nombre de valeurs dans un tableau

`array` **`array_count_values`** (`array` input)

`array_count_values` retourne un tableau contenant les valeurs du tableau input comme clés et leur fréquence comme valeur.

Exemple avec `array_count_values`

```
<?php
$array = array(1, "bonjour", 1, "monde", "bonjour");
print_r(array_count_values($array));
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [1] => 2
    [bonjour] => 2
    [monde] => 1
)
```

Voir aussi `count` , `array_unique` , `array_values` et `count_chars` .

8.3.12 `array_diff_assoc()` : Calcule la différence de deux tableaux, en prenant en compte les clés

`array_diff_assoc` (`array` `array1` , `array` `array2` , `array` ...)

`array_diff_assoc` retourne un tableau contenant les valeurs du tableau `array1` qui ne sont présentes dans aucun autre argument `array2` , Notez que les clés sont utilisées durant la comparaison, contrairement à `array_diff` .

Exemple avec `array_diff_assoc`

```
<?php
$array1 = array("a" => "green", "b" => "brown", "c" => "blue", "red");
$array2 = array("a" => "green", "yellow", "red");
$result = array_diff_assoc($array1, $array2);

print_r($result);
?>
```

Les occurrences multiples du tableau `array1` ne sont pas traités de la même façon. Le résultat est :

```
Array
(
    [b] => brown
    [c] => blue
    [0] => red
)
```

Dans notre exemple ci-dessus, vous pouvez voir que la paire "a" => "vert" est présente dans les deux tableaux, et donc, n'est pas présente dans le résultat de la fonction. Au contraire, la paire 0 => "rouge" est présente dans le résultat, car le second argument "rouge" possède une clé qui est 1 .

Les deux valeurs de la paire **clé => valeur** sont considérées égales uniquement si (string) `$elem1`

=== (string) \$elem2 . En d'autre termes, une comparaison stricte est faite sur les représentations des index, avec le type chaîne.

Note

Notez bien que cette fonction ne travaille que sur une dimension de tableau. Bien sur, vous pouvez utiliser des sous dimensions de tableau comme ceci : `array_diff_assoc($array1[0], $array2[0]);` .

Voir aussi [array_diff](#) , [array_intersect](#) et [array_intersect_assoc](#) .

8.3.13 array_diff_key() : Calcul la différence de deux tableaux en utilisant les clés pour comparaison

array **array_diff_key** (array array1 , array array2 , array ...)

[array_diff_key](#) retourne un tableau contenant toutes les valeurs de array1 où les clés de ce tableau ne sont présentes dans aucun des autres arguments. Notez que l'associativité est préservée. Cette fonction fonctionne de la même façon que la fonction [array_diff](#) excepté le fait que la comparaison est faite sur les clés au lieu des valeurs.

Exemple avec [array_diff_key](#)

```
<?php
$array1 = array('bleu' => 1, 'rouge' => 2, 'vert' => 3, 'violet' => 4);
$array2 = array('vert' => 5, 'bleu' => 6, 'jaune' => 7, 'cyan' => 8);

var_dump(array_diff_key($array1, $array2));
?>
```

L'exemple ci-dessus va afficher :

```
array(2) {
["rouge"]=>
int(2)
["violet"]=>
int(4)
}
```

Les deux clés depuis les paires clé => valeur sont considérées comme égales uniquement si (string) \$cle1 === (string) \$cle2 . En d'autres mots, une analyse du type stricte est exécuté, donc, le type doit être exactement le même.

Note

Notez que cette fonction vérifie uniquement une dimension d'un tableau possédant n dimensions. Bien sûr, vous pouvez vérifier une dimension plus profonde en utilisant, par exemple, `array_diff_key($array1[0], $array2[0]);` .

Voir aussi [array_diff](#) , [array_udiff](#) , [array_diff_assoc](#) , [array_diff_uassoc](#) , [array_udiff_assoc](#) , [array_udiff_uassoc](#) , [array_diff_ukey](#) , [array_intersect](#) , [array_intersect_assoc](#) , [array_intersect_uassoc](#) , [array_intersect_key](#) et [array_intersect_ukey](#) .

8.3.14 array_diff_uassoc() : Calcule la différence entre deux tableaux associatifs, à l'aide d'une fonction utilisateur

array **array_diff_uassoc** (array array1 , array array2 , array ... , callback key_compare_func)

array_diff_uassoc retourne un array contenant toutes les valeurs de array1 qui ne sont pas présentes dans l'un des autres arguments array2 , ... , etc. Notez que les clés sont utilisées dans les comparaisons, contrairement à array_diff .

Cette comparaison est effectuée avec la fonction callback key_compare_func . Cette fonction doit retourner un entier plus petit que, égal à, ou plus grand que zéro si le premier argument est respectivement plus petit que, égal au, ou plus grand que le second. Ce comportement est différent de celui de array_diff_assoc où une fonction interne de comparaison des index est utilisée.

Exemple avec array_diff_uassoc

```
<?php
function key_compare_func($a, $b) {
    if ($a === $b) return 0;
    return ($a > $b)? 1:-1;
}

$array1 = array("a" => "vert", "b" => "marron", "c" => "bleu", "rouge");
$array2 = array("a" => "vert", "jaune", "rouge");
$result = array_diff_uassoc($array1, $array2, "key_compare_func");

?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [b] => marron
    [c] => bleu
    [0] => rouge
)
```

Dans notre exemple, vous voyez que la paire "a" => "vert" est présente dans les deux tableaux et donc ne figure pas dans le résultat de la fonction. Contrairement à cela, la paire 0 => "rouge" est dans le résultat car dans le second argument, la clé de "rouge" est 1 .

L'égalité de deux indices est vérifiée par la fonction utilisateur.

Note

Notez que cette fonction ne vérifie qu'une seule dimension d'un tableau multi-dimensionnel. Vous pouvez, bien sûr, tester une dimension particulière en utilisant par exemple, array_diff_uassoc(\$array1[1], \$array2[1], "key_compare_func"); .

Voir aussi array_diff , array_diff_assoc , array_udiff , array_udiff_assoc , array_udiff_uassoc , array_intersect , array_intersect_assoc , array_uintersect , array_uintersect_assoc et array_uintersect_uassoc .

8.3.15 **array_diff_ukey()** : Calcule la différence entre deux tableaux en utilisant une fonction de callback sur les clés pour comparaison

array **array_diff_ukey** (array array1 , array array2 , array ... , callback key_compare_func)

array_diff_ukey retourne un tableau contenant toutes les valeurs de array1 où les clés de ce tableau ne sont présentes dans aucun des autres arguments array1 , Notez que l'associativité est préservée. Cette fonction fonctionne de la même façon que la fonction array_diff excepté le fait que la comparaison est faite sur les clés au lieu des valeurs.

Cette comparaison est effectuée en utilisant une fonction de callback fournie. Elle doit retourner un entier plus petit que, égal à ou plus grand que 0 si la première clé est considérée comme, respectivement, plus petite que, égale à ou plus grande que la seconde.

Exemple avec array_diff_ukey

```
<?php
function key_compare_func($key1, $key2)
{
    if ($key1 == $key2)
        return 0;
    else if ($key1 > $key2)
        return 1;
    else
        return -1;
}

$array1 = array('bleu' => 1, 'rouge' => 2, 'vert' => 3, 'violet' => 4);
$array2 = array('vert' => 5, 'bleu' => 6, 'jaune' => 7, 'cyan' => 8);

var_dump(array_diff_ukey($array1, $array2, 'key_compare_func'));
?>
```

L'exemple ci-dessus va afficher :

```
array(2) {
  ["rouge"]=>
  int(2)
  ["violet"]=>
  int(4)
}
```

Note

Notez que cette fonction vérifie uniquement une dimension d'un tableau possédant n dimensions. Bien sûr, vous pouvez vérifier une dimension plus profonde en utilisant, par exemple, array_diff_ukey(\$array1[0], \$array2[0], 'callback_func'); .

Voir aussi array_diff , array_udiff array_diff_assoc , array_diff_uassoc , array_udiff_assoc , array_udiff_uassoc , array_diff_key , array_intersect , array_intersect_assoc , array_intersect_uassoc , array_intersect_key et array_intersect_ukey .

8.3.16 array_diff() : Calcule la différence entre deux tableaux

array **array_diff** (array array1 , array array2 , array ...)

array_diff retourne un tableau qui contient toutes les valeurs du tableau array1 qui sont absentes de tous les autres arguments array2 , ... , etc. Notez que les clés sont préservées.

Exemple avec array_diff

```
<?php
$array1 = array ("a" => "vert", "rouge", "bleu", "rouge");
$array2 = array ("b" => "vert", "jaune", "rouge");
```

8.3.15 array_diff_ukey() : Calcule la différence entre deux tableaux en utilisant une fonction de callback sur

```
$result = array_diff ($array1, $array2);

print_r($result);
?>
```

Les valeurs multiples dans array1 seront toutes traitées de la même façon.

L'exemple ci-dessus va afficher :

```
Array
(
[1] => bleu
)
```

Note

Deux éléments sont considérés comme égaux si et seulement si (string) \$elem1 === (string) \$elem2 . En clair : lorsque la représentation en chaîne de caractères est identique.

Note

Notez que cette fonction ne vérifie qu'une dimension d'un tableau à n-dimensions. Bien sur, vous pouvez vérifier des dimensions plus profondes en utilisant `array_diff($array1[0], $array2[0]);` .

Attention

Cette fonction était inutilisable en PHP 4.0.4!

Voir aussi [array_diff_assoc](#) , [array_intersect](#) et [array_intersect_assoc](#) .

8.3.17 array_fill() : Remplis un tableau avec une même valeur

array **array_fill** (int start_index , int num , mixed value)

`array_fill` crée un tableau avec num entrées de la valeur value . Les index commençant à la valeur start_index . Notez que si num n'est pas un nombre supérieur à zéro, PHP affichera une alerte.

Exemple avec `array_fill`

```
<?php
$a = array_fill(5, 6, 'banane');
print_r($a);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[5] => banane
[6] => banane
[7] => banane
[8] => banane
[9] => banane
[10] => banane
)
```

Voir aussi [str_repeat](#) et [range](#) .

8.3.18 array_filter() : Filtre les éléments d'un tableau

array **array_filter** (array input , *callback* callback)

array_filter retourne un tableau contenant les éléments du tableau input , filtrés grâce à la fonction callback . Si input est un tableau associatif, les clés sont préservées.

Exemple avec array_filter

```
<?php
function impair($var)
{
    return ($var % 2 == 1);
}

function pair($var)
{
    return ($var % 2 == 0);
}

$array1 = array ("a"=>1, "b"=>2, "c"=>3, "d"=>4, "e"=>5);
$array2 = array (6, 7, 8, 9, 10, 11, 12);

echo "Impairs :\n";
print_r(array_filter($array1, "impair"));
echo "Pairs :\n";
print_r(array_filter($array2, "pair"));
?>
```

L'exemple ci-dessus va afficher :

Impairs :

```
Array
(
    [a] => 1
    [c] => 3
    [e] => 5
)
```

Pairs :

```
Array
(
    [0] => 6
    [2] => 8
    [4] => 10
    [6] => 12
)
```

La fonction appelée ne doit pas modifier le tableau lui-même, c'est à dire ajouter ou supprimer une valeur. Si des éléments sont modifiés, le comportement de array_filter est indéfini.

Si la fonction callback n'est pas fournie, array_filter va supprimer toutes les entrées de input qui sont égales à FALSE . Voyez conversion en booléen pour plus d'informations.

array_filter sans callback

```
<?php

$entry = array(
    0 => 'foo',
```

```

        1 => false,
        2 => -1,
        3 => null,
        4 => ''
    );

print_r(array_filter($entry));
?>

```

L'exemple ci-dessus va afficher :

```

Array
(
    [0] => foo
    [2] => -1
)

```

Voir aussi [array_map](#) , [array_reduce](#) et [array_walk](#) .

8.3.19 `array_flip()` : Remplace les clés par les valeurs, et les valeurs par les clés

`array` **array_flip** (`array` `trans`)

`array_flip` retourne un tableau dont les clés sont les valeurs du précédent tableau `trans` , et les valeurs sont les clés.

Notez bien que les valeurs de `trans` doivent être des clés valides, c'est à dire qu'elles doivent être des entiers ([entier](#)) ou des chaînes de caractères ([chaîne de caractères](#)). Une alerte sera émise si une valeur est d'un type qui ne convient pas et la paire en question **ne sera pas inversée** .

Si une valeur n'est pas unique, seule la dernière clé sera utilisée comme valeur, et toutes les autres seront perdues.

`array_flip` retourne FALSE en cas d'échec.

Exemple avec `array_flip`

```

<?php
$trans = array_flip($trans);
$original = strstr($str, $trans);
?>

```

Exemple avec `array_flip` : collision

```

<?php
$trans = array ("a" => 1, "b" => 1, "c" => 2);
$trans = array_flip($trans);
print_r($trans);
?>

```

maintenant `$trans` vaut :

```

Array

```

```
(
[1] => b
[2] => c
)
```

Voir aussi [array_values](#) , [array_keys](#) et [array_reverse](#) .

8.3.20 `array_intersect_assoc()` : Calcule l'intersection de deux tableaux avec des tests sur les index

`array_intersect_assoc` (array array1 , array array2 , array ...)

`array_intersect_assoc` retourne un tableau contenant toutes les valeurs de array1 qui sont aussi présentes dans tous les autres arguments array2 , ... , etc. Notez que les clés sont utilisées durant la comparaison, contrairement à `array_intersect` .

Exemple avec `array_intersect_assoc`

```
<?php
$array1 = array ("a" => "vert", "b" => "brun", "c" => "bleu", "rouge");
$array2 = array ("a" => "vert", "jaune", "rouge");
$result_array = array_intersect_assoc ($array1, $array2);
print_r($result_array);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[a] => vert
)
```

Dans notre exemple, vous pouvez voir que la paire "a" => "vert" est présente dans les deux tableaux, et donc placée dans le dernier tableau. La valeur rouge n'est pas retournée car dans \$array1 son index est 2 tandis que dans le tableau \$array2 , son index est 1 .

Les deux valeurs de la paire clé => valeur sont considérées égales uniquement si (string) \$elem1 === (string) \$elem2 . En d'autre termes, une comparaison stricte est faite sur les représentation des index, avec le type chaîne.

Voir aussi [array_intersect](#) , [array_uintersect_assoc](#) , [array_intersect_uassoc](#) , [array_uintersect_uassoc](#) , [array_diff](#) et [array_diff_assoc](#) .

8.3.21 `array_intersect_key()` : Calcul l'intersection de deux tableaux en utilisant les clés pour comparaison

`array_intersect_key` (array array1 , array array2 , array ...)

`array_intersect_key` retourne un tableau contenant toutes les valeurs du tableau array1 qui contiennent des clés présentes dans tous les arguments.

Exemple avec `array_intersect_key`

```
<?php
```

```
$array1 = array('bleu' => 1, 'rouge' => 2, 'vert' => 3, 'violet' => 4);
$array2 = array('vert' => 5, 'bleu' => 6, 'jaune' => 7, 'cyan' => 8);

var_dump(array_intersect_key($array1, $array2));
?>
```

L'exemple ci-dessus va afficher :

```
array(2) {
  ["bleu"]=>
  int(1)
  ["vert"]=>
  int(3)
}
```

Dans cet exemple, vous pouvez voir que seules les clés 'bleu' et 'vert' sont présentes dans les deux tableaux et donc, elles sont retournées. Notez également que les valeurs pour les clés 'bleu' et 'vert' diffèrent entre les deux tableaux. Néanmoins, elles correspondent toujours car uniquement les clés sont vérifiées. Les valeurs retournées sont celles du tableau array1 .

Les deux clés depuis les paires clé => valeur sont considérées comme égales uniquement si (string) \$cle1 === (string) \$cle2 . En d'autres mots, une analyse du type stricte est exécuté, donc, le type doit être exactement le même.

Voir aussi [array_diff](#) , [array_udiff](#) [array_diff_assoc](#) , [array_diff_uassoc](#) , [array_udiff_assoc](#) , [array_udiff_uassoc](#) , [array_diff_key](#) , [array_diff_ukey](#) , [array_intersect](#) , [array_intersect_assoc](#) , [array_intersect_uassoc](#) et [array_intersect_ukey](#) .

8.3.22 `array_intersect_uassoc()` : Calcul l'intersection de deux tableaux avec des tests sur les index, compare les index en utilisant une fonction de callback

```
array array_intersect_uassoc ( array array1 , array array2 , array ... , callback
key_compare_func )
```

[array_intersect_uassoc](#) retourne un tableau contenant toutes les valeurs du tableau array1 qui sont présentes dans tous les arguments. Notez que les clés sont utilisées dans la comparaison par opposition à la fonction [array_intersect](#) .

La comparaison d'index est effectuée en utilisant la fonction de callback fournie. Elle doit retourner un entier, plus petit que, égal à ou plus grand que zéro si le premier argument est considéré comme étant, respectivement, plus petit que, égal à ou plus grand le second.

Exemple avec `array_intersect_uassoc`

```
<?php
$array1 = array("a" => "vert", "b" => "marron", "c" => "bleu", "rouge");
$array2 = array("a" => "VERT", "B" => "marron", "jaune", "rouge");

print_r(array_intersect_uassoc($array1, $array2, "strcasecmp"));
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [b] => marron
)
```

Voir aussi [array_intersect](#) , [array_intersect_assoc](#) , [array_uintersect_assoc](#) , [array_uintersect_uassoc](#) , [array_intersect_key](#) et [array_intersect_ukey](#) .

8.3.23 [array_intersect_ukey\(\)](#) : Calcul l'intersection de deux tableaux en utilisant une fonction de callback sur les clés pour comparaison

array [array_intersect_ukey](#) (array array1 , array array2 , array ... , callback key_compare_func)

[array_intersect_ukey](#) retourne un tableau contenant toutes les valeurs du tableau array1 qui contiennent des clés présentent dans tous les arguments array2 ,

Cette comparaison est effectuée en utilisant une fonction de callback fournie par l'utilisateur. La fonction de callback doit retourner un entier plus petit que, égal à ou plus grand que 0 si la première clé est considérée, respectivement, comme plus petite que, égale à ou plus grande que la seconde.

Exemple avec [array_intersect_ukey](#)

```
<?php
function key_compare_func($key1, $key2)
{
    if ($key1 == $key2)
        return 0;
    else if ($key1 > $key2)
        return 1;
    else
        return -1;
}

$array1 = array('bleu' => 1, 'rouge' => 2, 'vert' => 3, 'violet' => 4);
$array2 = array('vert' => 5, 'bleu' => 6, 'jaune' => 7, 'cyan' => 8);

var_dump(array_intersect_ukey($array1, $array2, 'key_compare_func'));
?>
```

L'exemple ci-dessus va afficher :

```
array(2) {
    ["bleu"]=>
    int(1)
    ["vert"]=>
    int(3)
}
```

Dans cet exemple, vous pouvez voir que seules les clés 'bleu' et 'vert' sont présentes dans les deux tableaux et donc, elles sont retournées. Notez également que les valeurs pour les clés 'bleu' et 'vert' diffèrent entre les deux tableaux. Néanmoins, elles correspondent toujours car uniquement les clés sont vérifiées. Les valeurs retournées sont celles du tableau array1 .

Voir aussi [array_diff](#) , [array_udiff](#) [array_diff_assoc](#) , [array_diff_uassoc](#) , [array_udiff_assoc](#) ,

8.3.22 [array_intersect_uassoc\(\)](#) : Calcul l'intersection de deux tableaux avec des tests sur les indices pour comparaison

[array_udiff_uassoc](#) , [array_diff_key](#) , [array_diff_ukey](#) , [array_intersect](#) , [array_intersect_assoc](#) , [array_intersect_uassoc](#) et [array_intersect_key](#) .

8.3.24 [array_intersect\(\)](#) : Calcule l'intersection de tableaux

array **array_intersect** (array array1 , array array2 , array ...)

[array_intersect](#) retourne un tableau contenant toutes les valeurs de array1 qui sont présentes dans tous les autres arguments array2 , ... , etc. Notez que les clés sont préservées.

<p>Exemple avec array_intersect</p> <pre><?php \$array1 = array ("a" => "vert", "rouge", "bleu"); \$array2 = array ("b" => "vert", "jaune", "rouge"); \$result = array_intersect (\$array1, \$array2); print_r(\$result); ?></pre> <p>L'exemple ci-dessus va afficher :</p> <pre>Array ([a] => vert [0] => rouge)</pre> <p>Note</p> <p>Deux éléments sont considérés comme égaux si et seulement si (string) \$elem1 === (string) \$elem2 . En clair : lorsque la représentation en chaîne de caractères est identique.</p> <p>Voir aussi array_intersect_assoc , array_diff et array_diff_assoc .</p>
--

8.3.25 [array_key_exists\(\)](#) : Vérifie si une clé existe dans un tableau

bool **array_key_exists** (mixed key , array search)

[array_key_exists](#) retourne TRUE s'il existe une clé du nom de key dans le tableau search . key peut être n'importe quelle valeur valide d'index de tableau. [array_key_exists](#) fonctionne également sur les objets.

<p>Exemple avec array_key_exists</p> <pre><?php \$search_array = array('premier' => 1, 'second' => 4); if (array_key_exists('premier', \$search_array)) { echo "L'élément 'premier' existe dans le tableau"; } ?></pre> <p>Note</p> <p>Cette fonction s'appelait key_exists en PHP version 4.0.6.</p> <p>array_key_exists et isset</p>
--

`isset` ne retourne pas TRUE pour les clés de tableaux qui correspondent à une valeur NULL alors que c'est le cas pour `array_key_exists`.

```
<?php
$search_array = array('premier' => null, 'second' => 4);

// retourne false
isset($search_array['premier']);

// retourne true
array_key_exists('premier', $search_array);
?>
```

Voir aussi `isset`, `array_keys` et `in_array`.

8.3.26 `array_keys()` : Retourne toutes les clés d'un tableau

`array array_keys (array input , mixed search_value , bool strict)`

`array_keys` retourne les clés numériques et littérales du tableau input.

Si l'option `search_value` est spécifiée, seules les clés ayant cette valeur seront retournées. Sinon, toutes les clés de input sont retournées. Depuis PHP 5, vous pouvez utiliser le paramètre `strict` pour une comparaison incluant le type (===).

Exemple avec `array_keys`

```
<?php
$array = array (0 => 100, "couleur" => "rouge");
print_r(array_keys($array));

$array = array("bleu", "rouge", "vert", "bleu", "bleu");
print_r(array_keys($array, "bleu"));

$array = array("couleur" => array("bleu", "rouge", "vert"), "taille" => array("petit", "moyen", "grand"));
print_r(array_keys($array));
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [0] => 0
    [1] => couleur
)
Array
(
    [0] => 0
    [1] => 3
    [2] => 4
)
Array
(
    [0] => couleur
    [1] => taille
)
```

)

Voir aussi [array_values](#) et [array_key_exists](#) .

8.3.27 array_map() : Applique une fonction sur les éléments d'un tableau

array **array_map** (callback callback , array arr1 , array ...)

[array_map](#) retourne un tableau contenant tous les éléments du tableau arr1 , après leur avoir appliqué la fonction callback . Le nombre de paramètres de la fonction callback doit être égal au nombre de tableaux passés dans la fonction [array_map](#) , dans les arguments supplémentaires

Exemple avec [array_map](#)

```
<?php
function cube($n) {
    return $n*$n*$n;
}
$a = array(1, 2, 3, 4, 5);
$b = array_map("cube", $a);
print_r($b);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[0] => 1
[1] => 8
[2] => 27
[3] => 64
[4] => 125
)
```

[array_map](#) : utilisation de plusieurs tableaux

```
<?php
function parle_espagnol($n, $m) {
    return "Le nombre $n se dit $m en espagnol";
}

function map_espagnol($n, $m) {
    return array($n => $m);
}

$a = array(1, 2, 3, 4, 5);
$b = array("uno", "dos", "tres", "cuatro", "cinco");
$c = array_map("parle_espagnol", $a, $b);
print_r($c);
?>
```

L'exemple ci-dessus va afficher :

```
// Contenu de $c
Array
(
```

```

[0] => Le nombre 1 se dit uno en espagnol
[1] => Le nombre 2 se dit dos en espagnol
[2] => Le nombre 3 se dit tres en espagnol
[3] => Le nombre 4 se dit cuatro en espagnol
[4] => Le nombre 5 se dit cinco en espagnol
)

// Contenu de $d
Array
(
[0] => Array
(
[1] => uno
)

[1] => Array
(
[2] => dos
)

[2] => Array
(
[3] => tres
)

[3] => Array
(
[4] => cuatro
)

[4] => Array
(
[5] => cinco
)
)

```

Généralement, lors de l'utilisation de plusieurs tableaux, ils doivent être d'égale longueur, car la fonction de callback est appliquée de manière similaire à tous les tableaux. Si les tableaux sont de tailles inégales, les plus petits seront complétés avec des éléments vides.

Une utilisation intéressante de cette fonction est la construction de tableaux de tableaux, facilement réalisée en passant la valeur NULL comme nom de fonction de callback.

Création d'un tableau de tableaux

```

<?php
$a = array(1, 2, 3, 4, 5);
$b = array("one", "two", "three", "four", "five");
$c = array("uno", "dos", "tres", "cuatro", "cinco");
$d = array("un", "deux", "trois", "quatre", "cinq");

$d = array_map(null, $a, $b, $c, $d);
print_r($d);
?>

```

L'exemple ci-dessus va afficher :

```

Array
(
  [0] => Array
  (
    [0] => 1
    [1] => one
    [2] => uno
    [3] => un
  )

  [1] => Array
  (
    [0] => 2
    [1] => two
    [2] => dos
    [3] => deux
  )

  [2] => Array
  (
    [0] => 3
    [1] => three
    [2] => tres
    [3] => trois
  )

  [3] => Array
  (
    [0] => 4
    [1] => four
    [2] => cuatro
    [3] => quatre
  )

  [4] => Array
  (
    [0] => 5
    [1] => five
    [2] => cinco
    [3] => cinq
  )
)

```

Voir aussi [array_filter](#) , [array_reduce](#) , [array_walk](#) et information à propos de [callback](#) type.

8.3.28 `array_merge_recursive()` : Combine plusieurs tableaux ensemble, récursivement

`array array_merge_recursive (array array1 , array ...)`

`array_merge_recursive` rassemble les éléments de deux ou plusieurs tableaux ensemble, en ajoutant les éléments de l'un à la suite des éléments du précédent.

Si les tableaux passés en arguments ont les mêmes clés (chaînes de caractères), les valeurs sont alors rassemblées dans un tableau, de manière récursive, de façon à ce que, si l'une de ces valeurs est un tableau elle-même, la fonction la rassemblera avec les valeurs de l'entrée courante. Cependant, si deux tableaux ont la même clé numérique, la dernière valeur n'écrasera pas la précédente, mais sera ajoutée à la fin du tableau.

Exemple avec `array_merge_recursive`

```
<?php
  $ar1 = array("couleur" => array("favorie" => "rouge"), 5);
  $ar2 = array(10, "couleur" => array("favorie" => "vert", "rouge"));
  $result = array_merge_recursive($ar1, $ar2);
  print_r($result);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
  [couleur] => Array
  (
    [favorie] => Array
    (
      [0] => rouge
      [1] => vert
    )
  )
  [0] => rouge
)

[0] => 5
[1] => 10
)
```

Voir aussi [array_merge](#) .

8.3.29 `array_merge()` : Fusionne un ou plusieurs tableaux

`array array_merge (array array1 , array array2 , array ...)`

`array_merge` rassemble les éléments d'un ou de plusieurs tableaux `array1` , `array2` , ... ensemble, en ajoutant les valeurs de l'un à la fin de l'autre. Le résultat est un tableau.

Si les tableaux ont des clés en commun, la dernière valeur rencontrée écrasera l'ancienne. Pour les valeurs numériques, cela **n'arrive pas** , car alors, les valeurs sont ajoutées en fin de tableau.

Si vous passez un seul tableau à cette fonction et qu'il a des indices numériques, les clés seront réindexées normalement. Pour les tableaux associatifs, les entrées dupliquées seront fusionnées en continue.

Exemple avec `array_merge`

```
<?php
  $array1 = array ("couleur" => "rouge", 2, 4);
  $array2 = array ("a", "b", "couleur" => "vert", "forme" => "trapézoïde");
```

```
$result = array_merge ($array1, $array2);
print_r($result);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [couleur] => vert
    [0] => 2
    [1] => 4
    [2] => a
    [3] => b
    [forme] => trapézoÛde
)
```

Exemple avec array_merge

```
<?php
$array1 = array();
$array2 = array(1 => "data");
$result = array_merge($array1, $array2);
?>
```

N'oubliez pas que les index numériques seront renumérotés.

```
Array
(
    [0] => data
)
```

Si vous voulez préserver complètement les tableaux, et que vous souhaitez simplement les ajouter les uns aux autres, utilisez l'opérateur + :

```
<?php
$array1 = array();
$array2 = array(1 => "data");
$result = $array1 + $array2;
?>
```

Les index numériques seront préservés. D'où le résultat :

```
Array
(
    [1] => data
)
```

Note

Les clés communes seront écrasées successivement par les clés identiques suivantes.

Attention

Le comportement de array_merge a été modifié en PHP 5. Contrairement à PHP 4, array_merge n'accepte que des paramètres de type array . Cependant, vous pouvez utiliser le transtypage pour

fusionner les autres de types de valeurs. Voyez l'exemple ci-dessous.

Exemple avec `array_merge` en PHP 5

```
<?php
$beginning = 'foo';
$end = array(1 => 'bar');
$result = array_merge((array)$beginning, (array)$end);
print_r($result);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [0] => foo
    [1] => bar
)
```

Voir aussi [array_merge_recursive](#) , [array_combine](#) et [les opérateurs de tableaux](#) .

8.3.30 `array_multisort()` : Tri multi-dimensionnel de tableaux

`bool array_multisort (array ar1 , mixed arg , mixed ... , array ...)`

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

`array_multisort` sert à trier simultanément plusieurs tableaux, ou bien à trier un tableau multi-dimensionnel, suivant l'une ou l'autre de ses dimensions.

Les clés associatives ([chaîne de caractères](#)) seront maintenues, mais les clés numériques seront réindexées.

Les tableaux passés en arguments `ar1` , `arg` , ... sont traités comme les colonnes d'une table, triées par lignes (un peu comme la clause SQL ORDER BY). Le premier tableau est la clé primaire de tri. Les valeurs du premier tableau qui sont égales, sont triées grâce au tableau suivant, et ainsi de suite...

La structure des arguments de `array_multisort` est un peu inhabituelle, mais elle est plus souple. Le premier argument **doit** être un tableau. Par la suite, chaque argument peut être soit un tableau, soit une option de tri prise parmi les valeurs suivantes :

Options de tri :

- SORT_ASC : Tri en ordre ascendant
- SORT_DESC : Tri en ordre descendant

Options de type de tri :

- SORT_REGULAR : comparaison normale des valeurs
- SORT_NUMERIC : comparaison numérique des valeurs
- SORT_STRING : Comparaison alphabétique des valeurs

Une seule option de tri de chaque type peut être appliquée après un tableau. Une option ne s'applique qu'au tableau précédent. Tous les autres sont mis par défaut à SORT_ASC et SORT_REGULAR .

Trier plusieurs tableaux

```
<?php
$ar1 = array("10", 100, 100, "a");
$ar2 = array(1, 3, "2", 1);
array_multisort($ar1, $ar2);

var_dump($ar1);
var_dump($ar2);
?>
```

Dans cet exemple, après le tri, le premier tableau contient 10, "a", 100, 100. Le deuxième tableau contient 1, 1, "2", 3. Les entrées du second tableau correspondant aux valeurs jumelles du premier tableau (100 et 100), sont aussi triées.

```
array(4) {
[0]=> string(2) "10"
[1]=> string(1) "a"
[2]=> int(100)
[3]=> int(100)
}
array(4) {
[0]=> int(1)
[1]=> int(1)
[2]=> string(1) "2"
[3]=> int(3)
}
```

Classer un tableau multi-dimensionnel

```
<?php
$ar = array(
    array("10", 11, 100, 100, "a"),
    array( 1, 2, "2", 3, 1)
);
array_multisort($ar[0], SORT_ASC, SORT_STRING,
               $ar[1], SORT_NUMERIC, SORT_DESC);
var_dump($ar);
?>
```

Dans cet exemple, après le tri, le premier tableau contient 10, 100, 100, 11, "a" (tri alphabétique, ordre croissant); Le deuxième tableau contient 1, 3, "2", 2, 1 (tri numérique, ordre décroissant).

```
array(2) {
[0]=> array(5) {
[0]=> string(2) "10"
[1]=> int(100)
[2]=> int(100)
[3]=> int(11)
[4]=> string(1) "a"
}
}
```

```
[1]=> array(5) {
[0]=> int(1)
[1]=> int(3)
[2]=> string(1) "2"
[3]=> int(2)
[4]=> int(1)
}
}
```

Classer les résultats d'une base de données

Dans cet exemple, chaque élément du tableau data représente une ligne de la table. Ce type de données est typique d'un enregistrement de base de données.

Exemple de données :

```
volume | edition
-----+-----
67 | 2
86 | 1
85 | 6
98 | 2
86 | 6
67 | 7
```

Les données sont sous forme de tableau, appelé data . Cela est généralement issu, par exemple, par la fonction `mysql_fetch_assoc` .

```
<?php
$data[] = array('volume' => 67, 'edition' => 2);
$data[] = array('volume' => 86, 'edition' => 1);
$data[] = array('volume' => 85, 'edition' => 6);
$data[] = array('volume' => 98, 'edition' => 2);
$data[] = array('volume' => 86, 'edition' => 6);
$data[] = array('volume' => 67, 'edition' => 7);
?>
```

Dans cette exemple, nous allons trier la colonne volume par ordre décroissant, et la colonne edition par ordre croissant.

Nous avons un tableau de lignes, mais `array_multisort` nécessite un tableau de colonnes, donc nous utilisons le code suivant pour obtenir les colonnes et ainsi effectuer le tri.

```
<?php
// Obtient une liste de colonnes
foreach ($data as $key => $row) {
    $volume[$key] = $row['volume'];
    $edition[$key] = $row['edition'];
}

// Tri les données par volume décroissant, edition croissant
// Ajoute $data en tant que premier paramètre, pour trier par la clé commune
array_multisort($volume, SORT_DESC, $edition, SORT_ASC, $data);
?>
```

L'enregistrement est maintenant trié et ressemble à cela :

```

volume | edition
-----+-----
98 | 2
86 | 1
86 | 6
85 | 6
67 | 2
67 | 7

```

Tri non sensible à la casse

`SORT_STRING` et `SORT_REGULAR` sont sensibles à la casse, les chaînes de caractères commençant avec une lettre en majuscule viendront avant les chaînes de caractères commençant par une lettre en minuscule.

Pour effectuer une recherche insensible à la casse, effectuez le tri sur une copie minuscule des colonnes du tableau original.

```

<?php
$array = array('Alpha', 'atomic', 'Beta', 'bank');
$array_lowercase = array_map('strtolower', $array);

array_multisort($array_lowercase, SORT_ASC, SORT_STRING, $array);

print_r($array);
?>

```

L'exemple ci-dessus va afficher :

```

Array
(
    [0] => Alpha
    [1] => atomic
    [2] => bank
    [3] => Beta
)

```

8.3.31 `array_pad()` : Complète un tableau avec une valeur jusqu'à la longueur spécifiée

`array` **`array_pad`** (`array` `input`, `int` `pad_size`, `mixed` `pad_value`)

`array_pad` retourne une copie du tableau `input` complété jusqu'à la taille de `pad_size` avec la valeur `pad_value`. Si `pad_size` est positif, alors le tableau est complété à droite, s'il est négatif, il est complété à gauche. Si la valeur absolue de `pad_size` est plus petite que la taille du tableau `input`, alors le tableau n'est pas complété. Il est possible d'ajouter au maximum 1048576 d'un seul coup.

Exemple avec `array_pad`

```

<?php
$input = array(12, 10, 9);

$result = array_pad($input, 5, 0);
// Le résultat est array (12, 10, 9, 0, 0)

```

```

$result = array_pad($input, -7, -1);
// Le résultat est array (-1, -1, -1, -1, 12, 10, 9)

$result = array_pad($input, 2, "noop");
// pas complété
?>

```

Voir aussi [array_fill](#) et [range](#) .

8.3.32 array_pop() : Dépile un élément de la fin d'un tableau

mixed **array_pop** (array array)

[array_pop](#) dépile et retourne le dernier élément du tableau array , le raccourcissant d'un élément. Si array est vide, ou n'est pas un tableau, [array_pop](#) retourne NULL .

Note

Cette fonction remet le pointeur au début du tableau (équivalent de [reset](#)).

Exemple avec [array_pop](#)

```

<?php
$stack = array("orange", "banane", "pomme", "framboise");
$fruit = array_pop($stack);
print_r($stack);
echo $fruit . "\n";
?>

```

Après ceci, \$stack n'a plus que 3 éléments : "orange" , "banane" et "pomme" :

```

Array
(
    [0] => orange
    [1] => banane
    [2] => pomme
)
framboise

```

tandis que \$fruit contient "framboise" .

Voir aussi [array_push](#) , [array_shift](#) et [array_unshift](#) .

8.3.33 array_product() : Calcule le produit des valeurs du tableau

number **array_product** (array array)

[array_product](#) retourne le produit des valeurs du tableau array , sous forme d'un entier ou d'un nombre à virgule flottante.

Exemple avec [array_product](#)

```

$a = array(2, 4, 6, 8);
echo "produit(a) = " . array_product($a) . "\n";

```

L'exemple ci-dessus va afficher :

```
produit(a) = 384
```

8.3.34 `array_push()` : Empile un ou plusieurs éléments à la fin d'un tableau

```
int array_push ( array array , mixed var , mixed ... )
```

`array_push` considère `array` comme une pile, et empile les variables `var` , ... à la fin de `array` . La longueur du tableau `array` augmente d'autant. Cela a le même effet que

```
<?php
$array[] = $var;
?>
```

répété pour chaque `var` .

`array_push` retourne le nouveau nombre d'éléments du tableau.

Exemple avec `array_push`

```
<?php
$stack = array ("orange", "banane");
array_push ($stack, "pomme", "bleuet");
print_r($stack);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [0] => orange
    [1] => banane
    [2] => pomme
    [3] => bleuet
)
```

Note

Si vous utilisez la fonction `array_push` pour ajouter un élément à un tableau, il est mieux de la remplacer par l'opérateur `$array[] =` qui évite le passage par une fonction.

Note

`array_push` émettra une alerte si le premier argument n'est pas un tableau. Ceci diffère du comportement de `$var[]` où un nouveau tableau est créé.

Voir aussi `array_pop` , `array_shift` et `array_unshift` .

8.3.35 `array_rand()` : Prend une ou plusieurs valeurs, au hasard dans un tableau

```
mixed array_rand ( array input , int num_req )
```

[array_rand](#) est pratique lorsque vous voulez sélectionner une ou plusieurs valeurs au hasard dans un tableau. Le paramètre `input` est un tableau, et `num_req` spécifie le nombre de valeurs que vous voulez obtenir (par défaut, c'est 1).

Si vous ne demandez qu'une entrée, [array_rand](#) retourne l'index de la valeur. Sinon, elle retourne un tableau d'index. Cela vous permet de faire une sélection au hasard de clés, ou bien de valeurs.

Note

Depuis PHP 4.2.0, vous n'avez plus besoin d'initialiser le générateur de nombres aléatoires avec [srand](#) ou [mt_srand](#) car c'est fait automatiquement.

Exemple avec [array_rand](#)

```
<?php
srand ((double) microtime() * 10000000);
$input = array ("Neo", "Morpheus", "Trinitée", "Cypher", "Tank");
$rand_keys = array_rand ($input, 2);
print $input[$rand_keys[0]]."\n";
print $input[$rand_keys[1]]."\n";
?>
```

Voir aussi [shuffle](#) .

8.3.36 [array_reduce\(\)](#) : Réduit itérativement un tableau

mixed [array_reduce](#) (array `input` , callback `function` , int `initial`)

[array_reduce](#) applique itérativement la fonction `function` aux éléments du tableau `input` , de manière à réduire le tableau à une valeur simple. Si l'argument optionnel `initial` est disponible, il sera utilisé pour initialiser le processus, ou bien comme valeur finale si le tableau est vide. Si le tableau est vide et le paramètre `initial` n'est pas passé, [array_reduce](#) retourne NULL .

Exemple avec [array_reduce](#)

```
<?php
function rsum($v, $w) {
    $v += $w;
    return $v;
}
function rmul($v, $w) {
    $v *= $w;
    return $v;
}
$a = array(1, 2, 3, 4, 5);
$x = array();
$b = array_reduce($a, "rsum");
$c = array_reduce($a, "rmul", 10);
$d = array_reduce($x, "rsum", 1);
?>
```

Dans cet exemple, `$b` contiendra 15 , `$c` contiendra 1200 (= 10*1*2*3*4*5), et `$d` contiendra 1 .

Voir aussi [array_filter](#) , [array_map](#) , [array_unique](#) et [array_count_values](#) .

8.3.37 `array_reverse()` : Inverse l'ordre des éléments d'un tableau

`array` **`array_reverse`** (`array` `array` , `bool` `preserve_keys`)

`array_reverse` prend le tableau `array` et retourne un nouveau tableau qui contient les mêmes éléments mais dans l'ordre inverse, en préservant les clés si le paramètre `preserve_keys` vaut `TRUE` .

Exemple avec `array_reverse`

```

<?php
$input = array ("php", 4.0, array ("rouge", "vert"));
$result = array_reverse ($input);
$result_keyed = array_reverse ($input, TRUE);
print_r($result);
print_r($result_keyed);
?>

```

Ce code fait que `$result` et `$result_keyed` contiennent les mêmes éléments, mais qu'ils diffèrent au niveau des clés. `$result` et `$result_keyed` contiennent :

```

Array
(
  [0] => Array
  (
    [0] => rouge
    [1] => vert
  )
  [1] => 4
  [2] => php
)
Array
(
  [2] => Array
  (
    [0] => rouge
    [1] => vert
  )
  [1] => 4
  [0] => php
)

```

Note

Le second paramètre `preserve_keys` a été ajouté en PHP 4.0.3.

Voir aussi [array_flip](#) .

8.3.38 `array_search()` : Recherche dans un tableau la clé associée à une valeur

`mixed` **`array_search`** (`mixed` `needle` , `array` `haystack` , `bool` `strict`)

[array_search](#) recherche needle dans haystack et retourne la clé associée s'il la trouve, ou FALSE sinon.

Note

Si needle est une chaîne de caractères, la comparaison est effectuée du manière sensible à la casse.
--

Note

Avant PHP 4.2.0, array_search renvoyait NULL en cas d'échec, au lieu de FALSE .

Si le troisième paramètre strict vaut TRUE , alors array_search s'assurera aussi que le type de needle est le même que celui de la valeur trouvée dans haystack .

Si needle est trouvé plus d'une fois dans haystack , la première clé concordante est retournée. Pour trouver toutes les clés correspondantes, utilisez plutôt la fonction [array_keys](#) avec le paramètre optionnel search_value .

Exemple avec array_search

<pre><?php \$array = array(0 => "bleu", 1 => "rouge", 2 => "vert", 3 => "rouge"); \$key = array_search('vert', \$array); // \$key = 2; \$key = array_search('rouge', \$array); // \$key = 1; ?></pre>

Attention

Cette fonction peut retourner FALSE , mais elle peut aussi retourner une valeur équivalent à FALSE , utilisable dans une condition if simple. Utilisez l' <u>opérateur ===</u> pour tester la valeur de retour exacte de cette fonction.
--

Voir aussi [array_keys](#) , [array_values](#) , [array_key_exists](#) et [in_array](#) .

8.3.39 array_shift() : Dépèle un élément au début d'un tableau

mixed **array_shift** (array array)

[array_shift](#) extrait la première valeur d'un tableau et la retourne, en raccourcissant le tableau d'un élément, et en déplaçant tous les éléments vers le bas. Toutes les clés numériques seront modifiées pour commencer à zéro. Si array est vide, ou n'est pas un tableau, [array_shift](#) retourne NULL .

Note

Cette fonction remet le pointeur au début du tableau (équivalent de reset).
--

Exemple avec array_shift
--

<pre><?php \$stack = array("orange", "banane", "pomme", "bleuet"); \$fruit = array_shift(\$stack); print_r(\$stack); ?></pre>

\$stack contient encore trois éléments :
--

Array

```
(
[0] => banane
[1] => pomme
[2] => bleuet
)
```

et orange a été placé dans \$fruit .

Voir aussi [array_unshift](#) , [array_push](#) et [array_pop](#) .

8.3.40 array_slice() : Extrait une portion de tableau

array **array_slice** (array array , int offset , int length , bool preserve_keys)

[array_slice](#) retourne une série d'éléments du tableau array commençant à l'offset offset et représentant length éléments.

Si offset est non-négatif, la série commencera à cet offset dans le tableau array . Si offset est négatif, cette série commencera à l'offset offset mais en commençant à la fin du tableau array .

Si length est fourni et positif, alors la série retournée aura autant d'éléments. Si length est fourni et négatif, alors la série contiendra les éléments depuis l'offset offset jusqu'à length éléments en partant de la fin. Si length est omis, la séquence lira tous les éléments du tableau, depuis l'offset précisé jusqu'à la fin du tableau.

Notez que [array_slice](#) va renuméroter les index par défaut. Depuis PHP 5.0.2, vous pouvez modifier ce comportement en donnant la valeur de TRUE à preserve_keys .

Exemples avec [array_slice](#)

```
<?php
$input = array ("a", "b", "c", "d", "e");

$output = array_slice ($input, 2);      // retourne "c", "d" et "e"
$output = array_slice ($input, 2, -1); // retourne "c", "d"
$output = array_slice ($input, -2, 1); // retourne "d"
$output = array_slice ($input, 0, 3);  // retourne "a", "b" et "c"

// notez les clés d'index différentes
print_r(array_slice($input, 2, -1));
print_r(array_slice($input, 2, -1, true));
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[0] => c
[1] => d
)
Array
(
[2] => c
[3] => d
)
```

Voir aussi [array_splice](#) et [unset](#) .

8.3.41 `array_splice()` : Efface et remplace une portion de tableau

`array array_splice (array input , int offset , int length , array replacement)`

`array_splice` supprime les éléments désignés par `offset` et `length` du tableau `input` et les remplace par les éléments du tableau `replacement` , si ce dernier est présent. Cette fonction retourne un tableau contenant les éléments ainsi remplacés. Notez que les clés numériques de `input` ne sont pas préservées.

Si `offset` est positif, la série commencera à cet `offset` dans le tableau `input` . Si `offset` est négatif, cette série commencera à l'`offset` mais en commençant à la fin du tableau `input` .

Si `length` est donné et positif, alors la série aura autant d'éléments. Si `length` est donné et négatif, les éléments seront pris dans l'ordre inverse. Si `length` est omis, la séquence lira tous les éléments du tableau, depuis l'`offset` jusqu'à la fin du tableau. Conseil : pour supprimer tous les éléments du tableau depuis `offset` jusqu'à la fin, même si un tableau de remplacement `replacement` est spécifié, utilisez `count($input)` à la place de `length` .

Si `replacement` est précisé, alors les éléments supprimés sont remplacés par les éléments de ce tableau. Si l' `offset` et `length` sont tels que la taille du tableau ne change pas, alors les éléments du tableau de remplacement `replacement` sont insérés à partir de l'`offset` . si le tableau de remplacement `replacement` ne contient qu'un seul élément, il n'est pas obligatoire de forcer le type en tableau avec `array` , à moins que cette variable ne soit elle-même un tableau.

Les codes suivants sont équivalents :

<code>array_push(\$input, \$x, \$y)</code>	<code>array_splice(\$input, count(\$input), 0, array(\$x, \$y))</code>
<code>array_pop(\$input)</code>	<code>array_splice(\$input, -1)</code>
<code>array_shift(\$input)</code>	<code>array_splice(\$input, 0, 1)</code>
<code>array_unshift(\$input, \$x, \$y)</code>	<code>array_splice(\$input, 0, 0, array(\$x, \$y))</code>
<code>\$a[\$x] = \$y</code>	<code>array_splice(\$input, \$x, 1, \$y)</code>

`array_splice` retourne le tableau des éléments supprimés.

Exemples avec <code>array_splice</code>
<pre><?php \$input = array ("rouge", "vert", "bleu", "jaune"); array_splice (\$input, 2); // \$input vaut array ("rouge", "vert") \$input = array ("rouge", "vert", "bleu", "jaune"); array_splice (\$input, 1, -1); // \$input vaut array ("rouge", "jaune") \$input = array ("rouge", "vert", "bleu", "jaune"); array_splice (\$input, 1, count(\$input), "orange"); // \$input vaut array ("rouge", "orange") \$input = array ("rouge", "vert", "bleu", "jaune"); array_splice (\$input, -1, 1, array("black", "marron")); // \$input vaut array ("rouge", "vert", // "bleu", "black", "marron") ?></pre>

Voir aussi [array_slice](#) , [unset](#) et [array_merge](#) .

8.3.42 array_sum() : Calcule la somme des valeurs du tableau

number **array_sum** (array array)

[array_sum](#) retourne la somme des valeurs du tableau array , sous forme d'un entier ou d'un nombre à virgule flottante.

Exemple avec array_sum
<pre>\$a = array(2, 4, 6, 8); echo "sum(a) = " . array_sum(\$a) . "\n"; \$b = array("a"=>1.2, "b"=>2.3, "c"=>3.4); echo 'sum(b) = ' . array_sum(\$b) . "\n";</pre>
<p>L'exemple ci-dessus va afficher :</p>
<pre>sum(a) = 20 sum(b) = 6.9</pre>
Note
<p>Dans les versions de PHP antérieure à la version 4.2.1, cette fonction modifiait le tableau passé en argument et convertissait les chaînes de caractères en nombres (ce qui conduisait généralement à 0).</p>

8.3.43 array_udiff_assoc() : Calcule la différence entre des tableaux avec vérification des index, compare les données avec une fonction de callback

array **array_udiff_assoc** (array array1 , array array2 , array ... , callback data_compare_func)

[array_udiff_assoc](#) retourne un tableau contenant toutes les valeurs de array1 qui ne sont présentes dans aucun autre des arguments array2 , Notez que les clés sont utilisées dans les comparaisons contrairement à [array_diff](#) et [array_udiff](#) . La comparaison des données est effectuée en utilisant une fonction callback fournie par l'utilisateur, data_compare_func . Ce comportement est différent de celui de [array_diff_assoc](#) qui utilise une fonction de comparaison interne.

Exemple avec array_udiff_assoc
<pre><?php class cr { private \$priv_member; function cr(\$val) { \$this->priv_member = \$val; } static function comp_func_cr(\$a, \$b) { if (\$a->priv_member === \$b->priv_member) return 0; return (\$a->priv_member > \$b->priv_member)? 1:-1; } } \$a = array("0.1" => new cr(9), "0.5" => new cr(12), 0 => new cr(23), 1=> new cr(4), 2 => new cr(-1)); \$b = array("0.2" => new cr(9), "0.5" => new cr(22), 0 => new cr(3), 1=> new cr(4), 2 => new cr(-1));</pre>

```
$result = array_udiff_assoc($a, $b, array("cr", "comp_func_cr"));
print_r($result);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
  [0.1] => cr Object
  (
    [priv_member:private] => 9
  )
  [0.5] => cr Object
  (
    [priv_member:private] => 12
  )
  [0] => cr Object
  (
    [priv_member:private] => 23
  )
)
```

Dans notre exemple, vous voyez que la paire "1" => new cr(4) est présente dans les deux tableaux et donc absente du tableau résultant.

Pour la comparaison, la fonction callback de l'utilisateur est utilisée. Cette fonction doit retourner un entier inférieur, égal ou supérieur à zéro si le premier argument est respectivement plus petit, égal ou plus grand que le second.

Note

Notez que cette fonction ne vérifie qu'une seule dimension d'un tableau multi-dimensionnel. Vous pouvez, bien sûr, tester une dimension particulière en utilisant par exemple, `array_udiff_assoc($array1[1], $array2[1], "compare_func");` .

Voir aussi [array_diff](#) , [array_diff_assoc](#) , [array_diff_uassoc](#) , [array_udiff](#) , [array_udiff_uassoc](#) , [array_intersect](#) , [array_intersect_assoc](#) , [array_uintersect](#) , [array_uintersect_assoc](#) et [array_uintersect_uassoc](#) .

8.3.44 array_udiff_uassoc() : Calcule la différence de deux tableaux associatifs, compare les données et les index avec une fonction de callback

array **array_udiff_uassoc** (array array1 , array array2 , array ... , callback data_compare_func , callback key_compare_func)

[array_diff_uassoc](#) retourne un array contenant toutes les valeurs de array1 qui ne sont pas présentes dans l'un des autres arguments array2 , Notez que les clés sont utilisées dans les comparaisons contrairement à [array_diff](#) et [array_udiff](#) . Cette comparaison est effectuée par une fonction callback de l'utilisateur : data_compare_func . Ce comportement est différent de celui de [array_diff_assoc](#) qui utilise une fonction interne pour les comparaisons. La comparaisons des clés (indices) est effectuée par la fonction callback key_compare_func . Ce comportement est différent

8.3.43 array_udiff_assoc() : Calcule la différence entre des tableaux avec vérification des index, ~~600~~ compare le

de celui de `array_udiff_assoc`, vu que cette dernière utilise une fonction interne pour comparer les indices.

Exemple avec `array_udiff_uassoc`

```
<?php
class cr {
    private $priv_member;
    function cr($val) {
        $this->priv_member = $val;
    }
    static function comp_func_cr($a, $b) {
        if ($a->priv_member === $b->priv_member) return 0;
        return ($a->priv_member > $b->priv_member)? 1:-1;
    }
    static function comp_func_key($a, $b) {
        if ($a === $b) return 0;
        return ($a > $b)? 1:-1;
    }
}
$a = array("0.1" => new cr(9), "0.5" => new cr(12), 0 => new cr(23), 1=> new cr(4), 2 => new cr(-1));
$b = array("0.2" => new cr(9), "0.5" => new cr(22), 0 => new cr( 3), 1=> new cr(4), 2 => new cr(-1));

$result = array_udiff_uassoc($a, $b, array("cr", "comp_func_cr"), array("cr","comp_func_key"));
print_r($result);
?>
```

L'exemple ci-dessus va afficher :

Array

```
(
[0.1] => cr Object
(
[priv_member:private] => 9
)
)
[0.5] => cr Object
(
[priv_member:private] => 12
)
)
[0] => cr Object
(
[priv_member:private] => 23
)
)
```

Dans notre exemple, vous voyez que la paire "1" => new cr(4) est présente dans les deux tableaux et donc, absente du tableau résultant. Gardez en tête que vous devez fournir deux fonctions callback.

Pour la comparaison, la fonction callback de l'utilisateur est utilisée. Cette fonction doit retourner un entier inférieur, égal ou supérieur à zéro si le premier argument est respectivement plus petit, égal ou plus grand que le second.

Note

Notez que cette fonction ne vérifie qu'une seule dimension d'un tableau multi-dimensionnel. Vous pouvez, bien sûr, tester une dimension particulière en utilisant par exemple

```
array_udiff_uassoc($array1[0], $array2[0], "data_compare_func", "key_compare_func"); .
```

Voir aussi [array_diff](#) , [array_diff_assoc](#) , [array_diff_uassoc](#) , [array_udiff](#) , [array_udiff_assoc](#) , [array_intersect](#) , [array_intersect_assoc](#) , [array_uintersect](#) , [array_uintersect_assoc](#) et [array_uintersect_uassoc](#) .

8.3.45 array_udiff() : Calcule la différence entre deux tableaux en utilisant une fonction callback

array **array_udiff** (array array1 , array array2 , array ... , callback data_compare_func)

[array_udiff](#) retourne un tableau contenant toutes les valeurs de array1 qui ne sont pas présents dans les autres arguments array2 , Notez que les clés sont préservées. Pour la comparaison des données, la fonction data_compare_func est utilisée. Cette fonction doit retourner un entier plus petit que, égal à, ou plus grand que zéro si le premier argument est respectivement plus petit que, égal au, ou plus grand que le second. Ce comportement est différent de celui de [array_diff](#) où une fonction interne de comparaison des données est utilisée.

Exemple avec [array_udiff](#)

```
<?php
class cr {
    private $priv_member;
    function cr($val) {
        $this->priv_member = $val;
    }
    static function comp_func_cr($a, $b) {
        if ($a->priv_member === $b->priv_member) return 0;
        return ($a->priv_member > $b->priv_member)? 1:-1;
    }
}
$a = array("0.1" => new cr(9), "0.5" => new cr(12), 0 => new cr(23), 1=> new cr(4), 2 => new cr(-
$b = array("0.2" => new cr(9), "0.5" => new cr(22), 0 => new cr( 3), 1=> new cr(4), 2 => new cr(-

$result = array_udiff($a, $b, array("cr", "comp_func_cr"));
print_r($result);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [0.5] => cr Object
    (
        [priv_member:private] => 12
    )
    [0] => cr Object
    (
        [priv_member:private] => 23
    )
)
```

Note

Notez que cette fonction ne vérifie qu'une seule dimension d'un tableau multi-dimensionnel. Vous pouvez, bien sûr, tester une dimension particulière en utilisant par exemple `array_udiff($array1[0], $array2[0], "data_compare_func");` .

Voir aussi [array_diff](#) , [array_diff_assoc](#) , [array_diff_uassoc](#) , [array_udiff_assoc](#) , [array_udiff_uassoc](#) , [array_intersect](#) , [array_intersect_assoc](#) , [array_uintersect](#) , [array_uintersect_assoc](#) et [array_uintersect_uassoc](#) .

8.3.46 `array_uintersect_assoc()` : Calcul l'intersection de deux tableaux avec des tests sur l'index, compare les donnée en utilisant une fonction de callback

`array_uintersect_assoc` (`array` array1 , `array` array2 , `array` ... , `callback` data_compare_func)

`array_uintersect_assoc` retourne un tableau contenant toutes les valeurs du tableau array1 qui sont présentes dans tous les arguments. Notez que les clés sont utilisées dans la comparaison par opposition à la fonction `array_uintersect` . Les données sont comparées en utilisant une fonction de callback.

Exemple avec `array_uintersect_assoc`

```
<?php
$array1 = array("a" => "vert", "b" => "marron", "c" => "bleu", "rouge");
$array2 = array("a" => "VERT", "B" => "marron", "jaune", "rouge");

print_r(array_uintersect_assoc($array1, $array2, "strcasecmp"));
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [a] => vert
)
```

Pour la comparaison, une fonction de callback spécifiée par l'utilisateur est utilisée. Elle doit retourner un entier, plus petit que, égal à ou plus grand que zéro si le premier argument est considéré comme étant, respectivement, plus petit que, égal à ou plus grand le second.

Voir aussi [array_uintersect](#) , [array_intersect_assoc](#) , [array_intersect_uassoc](#) et [array_uintersect_uassoc](#) .

8.3.47 `array_uintersect_uassoc()` : Calcul l'intersection de deux tableaux avec des tests sur l'index, compare les données et les indexes des deux tableaux en utilisant une fonction de callback

`array_uintersect_uassoc` (`array` array1 , `array` array2 , `array` ... , `callback` data_compare_func , `callback` key_compare_func)

`array_uintersect_uassoc` retourne une tableau contenant toutes les valeurs du tableau array1 qui sont présentes dans tous les arguments. Notez que les clés sont utilisées dans la comparaison en

opposition à la fonction [array_uintersect](#) . Les données et les indexes sont comparés en utilisant une fonction de callback.

Exemple avec [array_uintersect_uassoc](#)

```
<?php
$array1 = array("a" => "vert", "b" => "marron", "c" => "bleu", "rouge");
$array2 = array("a" => "VERT", "B" => "marron", "jaune", "rouge");

print_r(array_uintersect_uassoc($array1, $array2, "strcasecmp", "strcasecmp"));
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
  [a] => vert
  [b] => marron
)
```

Pour la comparaison, une fonction de callback spécifiée par l'utilisateur est utilisée. Elle doit retourner un entier, plus petit que, égal à ou plus grand que zéro si le premier argument est considéré comme étant, respectivement, plus petit que, égal à ou plus grand le second.

Voir aussi [array_uintersect](#) , [array_intersect_assoc](#) , [array_intersect_uassoc](#) et [array_uintersect_assoc](#) .

8.3.48 [array_uintersect\(\)](#) : Calcul l'intersection de deux tableaux, compare les données en utilisant une fonction de callback

array [array_uintersect](#) (array array1 , array array2 , array ... , callback data_compare_func)

[array_uintersect](#) retourne un tableau contenant toutes les valeurs du tableau array1 qui sont présentes dans tous les arguments. Les données sont comparées en utilisant une fonction de callback.

Exemple avec [array_uintersect](#)

```
<?php
$array1 = array("a" => "vert", "b" => "marron", "c" => "bleu", "rouge");
$array2 = array("a" => "VERT", "B" => "marron", "jaune", "rouge");

print_r(array_uintersect($array1, $array2, "strcasecmp"));
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
  [a] => vert
  [b] => marron
  [0] => rouge
)
```

Pour la comparaison, une fonction de callback spécifiée par l'utilisateur est utilisée. Elle doit retourner un entier, plus petit que, égal à ou plus grand que zéro si le premier argument est considéré comme étant, respectivement, plus petit que, égal à ou plus grand le second.

Voir aussi [array_intersect](#) , [array_uintersect_assoc](#) , [array_intersect_uassoc](#) et [array_uintersect_uassoc](#) .

8.3.49 array_unique() : Dédoublonne un tableau

array **array_unique** (array array)

[array_unique](#) prend le tableau array et retourne un nouveau tableau, complètement dédoublonné.

Notez que les clés sont préservées. [array_unique](#) trie les valeurs traitées comme des chaînes dans un premier temps, puis garde la première clé rencontrée pour chaque valeur et ignore les clés suivantes. Cela ne signifie pas que la clé de la première valeur rencontrée du tableau array non trié sera conservé.

Note
Deux éléments sont considérés comme égaux si et seulement si (string) \$elem1 === (string) \$elem2 . En clair : lorsque la représentation en chaîne de caractères est identique.
Le premier élément sera utilisé.

Exemple avec array_unique
<pre><?php \$input = array ("a" => "vert", "rouge", "b" => "vert", "bleu", "rouge"); \$result = array_unique (\$input); print_r(\$result); ?></pre> <p>L'exemple ci-dessus va afficher :</p> <pre>Array ([a] => vert [0] => rouge [1] => bleu)</pre>

array_unique et les types de valeurs
<pre><?php \$input = array (4, "4", "3", 4, 3, "3"); \$result = array_unique (\$input); var_dump(\$result); ?></pre> <p>Le résultat du programme sera (PHP 4.0.6) :</p> <pre>array(2) {</pre>

```
[3]=>
int(4)
[4]=>
int(3)
}
```

8.3.50 `array_unshift()` : Empile un ou plusieurs éléments au début d'un tableau

int `array_unshift` (array `array` , mixed `var` , *mixed* ...)

`array_unshift` ajoute les éléments `var` , ... , passés en argument au début du tableau `array` . Notez que les éléments sont ajoutés comme un tout, et qu'ils restent dans le même ordre.

`array_unshift` retourne le nouveau nombre d'éléments du tableau `array` .

Exemples avec `array_unshift`

```
<?php
$queue = array ("orange", "banane");
array_unshift ($queue, "pomme", "framboise");
print_r($queue);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[0] => pomme
[1] => framboise
[2] => orange
[3] => banane
)
```

Voir aussi `array_shift` , `array_push` et `array_pop` .

8.3.51 `array_values()` : Retourne les valeurs d'un tableau

array `array_values` (array `input`)

`array_values` retourne les valeurs du tableau `input` et indexe numériquement le tableau.

Exemples avec `array_values`

```
<?php
$array = array ("taille" => "XL", "couleur" => "vert");
print_r(array_values ($array));
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
```

```
[0] => XL
[1] => vert
)
```

Voir aussi [array_keys](#) .

8.3.52 `array_walk_recursive()` : Applique une fonction utilisateur récursivement à chaque membre du tableau

`bool array_walk_recursive (array input , callback funcname , mixed userdata)`

`array_walk_recursive` applique la fonction utilisateur `funcname` à chaque élément du tableau `input` . Cette fonction se reproduira dans toutes les profondeurs du tableau. La valeur du paramètre `input` étant le premier et la clé/index, le second. Si le paramètre optionnel `userdata` est fourni, il sera passé comme troisième paramètre à la fonction de callback `funcname` .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Si `funcname` doit être exécuté avec les valeurs actuelles du tableau, spécifiez le troisième paramètre de `funcname` par [référence](#) . Alors, tous changements effectués sur les éléments de ce tableau seront également effectués sur le tableau original.

Exemple avec `array_walk_recursive`

```
<?php
$sweet = array('a' => 'apple', 'b' => 'banana');
$fruits = array('sweet' => $sweet, 'sour' => 'lemon');

function test_print($item, $key)
{
    echo "$key tient $item\n";
}

array_walk_recursive($fruits, 'test_print');
?>
```

L'exemple ci-dessus va afficher :

```
a tient apple
b tient banana
sour tient lemon
```

Vous devez noter que la clé 'sweet' n'est jamais affichée. Toute clé qui tient un [tableau](#) n'est pas passée à la fonction.

Voir aussi [array_walk](#) et information à propos de [callback](#) type.

8.3.53 `array_walk()` : Exécute une fonction sur chacun des éléments d'un tableau

`bool array_walk (array array , callback funcname , mixed userdata)`

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.3.51 `array_values()` : Retourne les valeurs d'un tableau

Exécute la fonction funcname définie par l'utilisateur sur chaque élément du tableau array . Typiquement, funcname prend deux paramètres. La valeur du paramètre input étant le premier et la clé/index, le second. Si le paramètre optionnel userdata est fourni, il sera passé comme troisième paramètre à la fonction définie par l'utilisateur funcname .

Si fonction requiert plus de paramètres que ceux donnés, une alerte `E_WARNING` sera générée à chaque fois que la fonction `array_walk` appellera funcname . Ces alertes peuvent ne pas être affichées en utilisant l'opérateur d'erreur PHP `@` lors de l'appel de la fonction `array_walk` ou en utilisant `error_reporting` .

Note

Si funcname doit travailler avec les véritables valeurs du tableau, spécifiez que le premier paramètre de funcname doit être passé par référence. Alors, les éléments seront directement modifiés dans le tableau.

Note

Passer les clés et userdata à funcname a été ajouté en PHP 4.0.0

`array_walk` n'est pas affecté par le pointeur interne du tableau array . `array_walk` traversera le tableau en totalité sans se soucier de la position du pointeur.

Les utilisateurs ne peuvent pas modifier le tableau lui-même depuis la fonction de callback. Par exemple, Ajout/Effacement d'éléments, réinitialisation d'éléments, etc. Si le tableau sur lequel `array_walk` est appliqué est changé, le comportement de la fonction est indéfini et non prévisible.

Exemple avec `array_walk`

```
<?php;
$fruits = array ("d"=>"citron", "a"=>"orange", "b"=>"banane", "c"=>"pomme");

function test_alter(&$item1, $key, $prefix) {
    $item1 = "$prefix: $item1";
}

function test_print ($item2, $key) {
    echo "$key. $item2<br />\n";
}

echo 'Avant... '."\n";
array_walk ($fruits, 'test_print');

array_walk ($fruits, 'test_alter', 'fruit');
echo '...et après :'. "\n";

array_walk ($fruits, 'test_print');
?>
```

L'exemple ci-dessus va afficher :

```
Avant...
d. citron
a. orange
b. banane
c. pomme
...et après :
d. fruit: citron
a. fruit: orange
b. fruit: banane
```

c. fruit: pomme

Voir aussi [array_walk_recursive](#) , [create_function](#) , [list](#) , [foreach](#) , [each](#) , [call_user_func_array](#) , [array_map](#) et information à propos de [callback](#) type.

8.3.54 array() : Crée un tableau

array **array** (*mixed* ...)

[array](#) retourne un tableau créé avec les paramètres passés On peut attribuer un index particulier à une valeur avec l'opérateur => .

Note

[array](#) est un élément de langage utilisé pour représenter des tableaux littéraux, et non pas une fonction au sens strict du terme.

La syntaxe "index => valeur", séparée par des virgules, définit les index et leur valeur. Un index peut être une chaîne ou un nombre. Si l'index est omis, un index numérique sera automatiquement généré (commençant à 0). Si l'index est un entier, le prochain index généré prendra la valeur d'index la plus grande + 1. Notez que si deux index identiques sont définis, le dernier remplacera le premier.

Avoir une virgule après avoir défini la dernière entrée, bien qu'inutile, est une syntaxe valide.

L'exemple suivant montre comment créer un tableau à deux dimensions, comment spécifier les index d'un tableau associatif, et comment générer automatiquement des index numériques.

Exemple avec [array](#)

```
<?php
$fruits = array (
    "fruits" => array ("a"=>"orange", "b"=>"banane", "c"=>"pomme"),
    "nombres" => array (1, 2, 3, 4, 5, 6),
    "trous" => array ("premier", 5 => "second", "troisième")
);
?>
```

Index automatique d'un tableau avec [array](#)

```
<?php
$array = array( 1, 1, 1, 1, 1, 8=>1, 4=>1, 19, 3=>13);
print_r($array);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [0] => 1
    [1] => 1
    [2] => 1
    [3] => 13
    [4] => 1
    [8] => 1
    [9] => 19
```

)

Notez bien que l'index '3' est défini deux fois, et conserve finalement sa dernière valeur de 13. L'index '4' est défini après l'index '8', et l'index généré suivant (valeur 19) est 9, puisque le plus grand index est alors 8.

Cet exemple crée un tableau dont les index commencent à 1.

Tableau d'index commençant à 1

```
<?php
$firstquarter = array(1 => 'Janvier', 'Février', 'Mars');
print_r($firstquarter);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [1] => Janvier
    [2] => Février
    [3] => Mars
)
```

Tout comme en Perl, vous pouvez accéder à une valeur d'un tableau dans des doubles guillemets. Cependant, avec PHP, vous devez entourer votre tableau avec des parenthèses.

Accéder à un tableau dans des doubles guillemets

```
<?php
$foo = array('bar' => 'baz');
echo "Hello {$foo['bar']}!"; // Hello baz!
?>
```

Voir aussi [array_pad](#) , [list](#) , [count](#) , [foreach](#) et [range](#) .

8.3.55 arsort() : Trie un tableau en ordre inverse

bool **arsort** (array array , int sort_flags)

[arsort](#) trie le tableau array de telle manière que la corrélation entre les index et les valeurs soit conservée. L'usage principal est lors de tri de tableaux associatifs où l'ordre des éléments est important.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec [arsort](#)

```
<?php
$fruits = array("d" => "lemon", "a" => "orange", "b" => "banana", "c" => "apple");
arsort($fruits);
foreach ($fruits as $key => $val) {
    echo "$key = $val\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
a = orange
d = lemon
b = banana
c = apple
```

Les fruits ont été triés en ordre alphabétique inverse, et leurs index respectifs ont été conservés.

Vous pouvez modifier le comportement de cette fonction en utilisant le paramètre optionnel `sort_flags` . Pour plus de détails, voyez la fonction [sort](#) .

Voir aussi [asort](#) , [rsort](#) , [ksort](#) et [sort](#) .

8.3.56 asort() : Trie un tableau et conserve l'association des index

bool **asort** (array array , int sort_flags)

[asort](#) trie le tableau array de telle manière que la corrélation entre les index et les valeurs soit conservée. L'usage principal est lors de tri de tableaux associatifs où l'ordre des éléments est important.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec [asort](#)

```
<?php
$fruits = array("d"=>"papaye",
               "a"=>"orange",
               "b"=>"banane",
               "c"=>"ananas");
asort($fruits);
foreach ($fruits as $key => $val) {
    echo "$key = $val\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
fruits[c] = ananas
fruits[b] = banane
fruits[a] = orange
fruits[d] = papaye
```

Les fruits ont été triés par ordre alphabétique, et leurs index respectifs ont été conservés.

Vous pouvez modifier le comportement de cette fonction en utilisant le paramètre optionnel `sort_flags` . Pour plus de détails, voyez le manuel pour la fonction [sort](#) .

Voir aussi [arsort](#) , [rsort](#) , [ksort](#) et [sort](#) .

8.3.57 compact() : Crée un tableau à partir de variables et de leur valeur

array **compact** (mixed varname , *mixed* ...)

compact accepte différents paramètres varname . Les paramètres peuvent être des variables contenant des chaînes, ou un tableau de chaînes, qui peut contenir d'autres tableaux de noms de variables, que compact traitera récursivement.

Pour chacun des arguments varname , ... , compact recherche une variable avec un même nom dans la table courante des symboles, et l'ajoute dans le tableau, de manière à avoir la relation nom => 'valeur de variable'. En bref, c'est le contraire de la fonction extract . compact retourne le tableau ainsi créé.

Toute chaîne non reconnue dans la table des symboles sera tout simplement ignorée.

Note
Gotcha
Parce que les <u>variables variables</u> ne doivent pas être utilisées avec les <u>tableaux Superglobaux</u> dans des fonctions, les tableaux Superglobaux ne doivent pas être passés dans la fonction <u>compact</u> .

Exemple avec <u>compact</u>
<pre><?php \$ville = "Montréal"; \$province = "Québec"; \$evenement = "Conférence PHP Québec"; \$location_vars = array("ville", "province"); \$result = compact("evenement", "rien ici", \$location_vars); print_r(\$result); ?></pre>
L'exemple ci-dessus va afficher :
<pre>Array ([evenement] => Conférence PHP Québec [ville] => Montréal [province] => Québec)</pre>

Voir aussi extract .

8.3.58 count() : Compte le nombre d'éléments d'un tableau ou le nombre de propriétés d'un objet

int **count** (mixed var , *int* mode)

count retourne le nombre d'éléments dans var , qui est généralement un tableau et tout le reste, sauf les objets, n'aura qu'un élément.

Pour les objets, count retourne le nombre de propriétés non-statiques, sans tenir compte de la

visibilité. Si vous avez [SPL](#) d'installé, vous pouvez vous accrocher à la fonction [count](#) en implémentant l'interface [Countable](#) . Cette interface a exactement une méthode, [count](#) , qui retourne la valeur retournée par la fonction [count](#) .

Si le paramètre var n'est ni un objet, ni un tableau, 1 sera retourné. Il y a néanmoins une exception : si le paramètre var vaut NULL , 0 sera retourné.

Note

Le paramètre optionnel mode est disponible depuis PHP 4.2.0.

Si le paramètre optionnel mode vaut COUNT_RECURSIVE (ou 1), [count](#) va compter récursivement les tableaux. C'est particulièrement pratique pour compter le nombre d'éléments d'un tableau. La valeur par défaut pour mode est 0 . [count](#) ne détecte pas la récursion infinie.

Attention

[count](#) peut retourner 0 pour une variable qui n'a pas été affectée, ou pour un tableau vide. Utilisez plutôt [isset](#) pour tester si la variable existe.

Reportez-vous à la section sur les [Tableaux](#) du manuel, pour plus de détails sur le fonctionnement des tableaux en PHP.

Exemple avec [count](#)

```
<?php
$a[0] = 1;
$a[1] = 3;
$a[2] = 5;
$result = count ($a);
// $result == 3

$b[0] = 7;
$b[5] = 9;
$b[10] = 11;
$result = count ($b);
// $result == 3;

$result = count(null);
// $result == 0;

$result = count(false);
// $result == 1;

$obj = new stdClass;
$obj->foo = 'Une propriété';
$obj->bar = 'Une autre propriété';
$result = count($obj);
// $result == 2;
?>
```

Exemple récursif avec [count](#) (PHP >= 4.2.0)

```
<?php
$food = array( 'fruits' => array('orange', 'banane', 'pommes'),
              'legumes' => array('carotte', 'chou','pois'));

// count récursif
echo count($food, COUNT_RECURSIVE); // affiche 8

// count normal
echo count($food); // affiche 2

?>
```

8.3.58 count() : Compte le nombre d'éléments d'un tableau ou le nombre de propriétés d'un objet

Voir aussi [is_array](#) , [isset](#) et [strlen](#) .

8.3.59 current() : Retourne l'élément courant du tableau

mixed **current** (array array)

Chaque tableau entretient un pointeur interne, qui est initialisé lorsque le premier élément est inséré dans le tableau.

[current](#) ne fait que retourner l'élément courant pointé par le pointeur interne du tableau array . [current](#) ne déplace pas le pointeur. Si le pointeur est au-delà du dernier élément de la liste, [current](#) retourne FALSE .

Attention

Si le tableau contient des éléments vides ou des zéros (0 ou "", la chaîne vide) alors [current](#) retournera FALSE pour ces éléments. Il est donc impossible de déterminer si vous êtes réellement à la fin de la liste en utilisant la fonction [current](#) . Pour passer en revue proprement un tableau qui peut contenir des éléments vides ou des zéros, utilisez la fonction [each](#) .

Exemple avec [current](#)

```
<?php
$transport = array('pied', 'velo', 'voiture', 'avion');
$mode = current($transport); // $mode = 'pied';
$mode = next($transport);    // $mode = 'velo';
$mode = current($transport); // $mode = 'velo';
$mode = prev($transport);    // $mode = 'pied';
$mode = end($transport);     // $mode = 'avion';
$mode = current($transport); // $mode = 'avion';
?>
```

Voir aussi [end](#) , [key](#) , [next](#) , [prev](#) et [reset](#) .

8.3.60 each() : Retourne chaque paire clé/valeur d'un tableau

array **each** (array array)

[each](#) retourne la paire clé/valeur courante du tableau array et avance le pointeur de tableau. Cette paire est retournée dans un tableau de 4 éléments, avec les clés 0 , 1 , key , et value . Les éléments 0 et key contiennent le nom de la clé et 1 et value contiennent la valeur.

Si le pointeur interne de tableau est au-delà de la fin du tableau, [each](#) retourne FALSE .

Exemples avec [each](#)

```
<?php
$foo = array("ghislain", "damien", "didou", "guillaume", "jim", "christophe");
$bar = each($foo);
print_r($bar);
?>
```

L'exemple ci-dessus va afficher :

Array

```
(
[1] => ghislain
[value] => ghislain
[0] => 0
[key] => 0
)

<?php
$foo = array("Mehdi" => "Didou", "Damien" => "Dams");
$bar = each($foo);
print_r($bar);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[1] => Didou
[value] => Didou
[0] => Mehdi
[key] => Mehdi
)
```

each est typiquement utilisée en conjonction avec list pour passer en revue un tableau. Par exemple :

Passer en revue un tableau avec each

```
<?php
$fruit = array('a' => 'pomme', 'b' => 'banane', 'c' => 'canneberge');

reset($fruit);
while (list($key, $val) = each($fruit)) {
    echo "$key => $val\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
a => pomme
b => banane
c => canneberge
```

Après chaque appel à each, le pointeur de tableau est déplacé au prochain élément, ou sur le dernier élément, lorsqu'on arrive à la fin. Vous devez utiliser reset si vous voulez traverser le tableau à nouveau avec each.

Attention

Assigner un tableau à une autre variable remet le pointeur du tableau original à zéro. A cause de ce comportement, nous aurions pu provoquer une boucle infinie dans notre exemple si nous avions assigné \$fruit à une autre variable dans notre boucle.

Voir aussi key, list, current, reset, next, prev et foreach.

8.3.61 end() : Positionne le pointeur de tableau en fin de tableau

mixed **end** (array array)

end déplace le pointeur interne du tableau array jusqu'au dernier élément et retourne sa valeur.

Exemple avec end

```
<?php
$fruits = array('pomme', 'banane', 'myrtille');
echo end($fruits); // myrtille
?>
```

Voir aussi current , each , prev , next et reset .

8.3.62 extract() : Importe les variables dans la table des symboles

int **extract** (array var_array , int extract_type , string prefix)

extract sert à exporter un tableau vers la table des symboles. Elle prend un tableau associatif var_array , crée les variables dont les noms sont les index de ce tableau, et leur affecte la valeur associée. Pour chaque paire clé/valeur, extract crée une variable, avec les paramètres extract_type et prefix .

Note

Depuis la version 4.0.5, extract retourne le nombre de variables extraites.

Note

EXTR_IF_EXISTS et EXTR_PREFIX_IF_EXISTS ont été introduites en PHP version 4.2.0.

Note

EXTR_REFS a été introduite en PHP 4.3.0.

extract vérifie l'existence de la variable avant de la créer. Le traitement des collisions est déterminé par extract_type . Ce paramètre peut prendre une des valeurs suivantes :

EXTR_OVERWRITE

Lors d'une collision, réécrire la variable existante.

EXTR_SKIP

Lors d'une collision, ne pas réécrire la variable existante.

EXTR_PREFIX_SAME

Lors d'une collision, ajouter le préfixe prefix , et créer une nouvelle variable.

EXTR_PREFIX_ALL

Ajouter le préfixe prefix pour toutes les variables. A partir de la version 4.0.5, les variables numériques sont aussi préfixées.

EXTR_PREFIX_INVALID

Préfixer uniquement les variables aux noms invalides ou numériques avec le préfixe prefix . Ceci a été ajouté en PHP 4.0.5.

EXTR_IF_EXISTS

Écrase la variable uniquement si elle existe déjà dans la table des symboles, et sinon, ne rien faire. Ceci est pratique pour définir une liste de variables valides, puis d'extraire du tableau les valeurs que vous avez déjà définies grâce à \$_REQUEST , par exemple. Cette option a été ajoutée en PHP 4.2.0.

EXTR_PREFIX_IF_EXISTS

Ne crée que des variables préfixées, si la version non préfixée de la même variable existe dans la table des symboles courante. Cette option a été ajoutée en PHP 4.2.0.

EXTR_REFS

Extrait les variables sous forme de références. Cela signifie que les valeurs des variables importées font toujours référence aux valeurs des variables d'origine dans le paramètre `var_array`. Vous pouvez utiliser cette option seule, ou bien la combiner avec d'autres options avec l'opérateur OR, dans le paramètre `extract_type`. Cette option a été ajoutée en PHP 4.3.0.

Si `extract_type` est omis, `extract` utilise `EXTR_OVERWRITE` par défaut.

Notez que prefix n'est nécessaire que pour les valeurs de `extract_type` suivantes :

`EXTR_PREFIX_SAME`, `EXTR_PREFIX_ALL`, `EXTR_PREFIX_INVALID` ou `EXTR_PREFIX_IF_EXISTS`. Si le résultat préfixé n'est pas un nom de variable valide, il ne sera pas importé dans la table des symboles. Les préfixes sont automatiquement séparés de l'index du tableau par un caractère de soulignement.

`extract` retourne le nombre de variables réellement importées dans la table des symboles.

Attention

N'utilisez pas `extract` sur des données inconnues, comme les données utilisateurs (`$_GET`, etc). Si vous le faites, par exemple, pour rendre compatible un vieux code avec `register_globals` à Off de façon temporaire, assurez-vous d'utiliser l'une des constantes `extract_type` qui n'écrasent pas les valeurs, comme `EXTR_SKIP`. Sachez aussi que vous devez maintenant extraire dans le même ordre comme il est défini dans `variables_order` dans `php.ini`.

Une utilisation possible de la fonction `extract` est l'exportation vers la table des symboles de tableaux de variables retournés par `wddx_deserialize`.

Exemple avec `extract`

```
<?php
// Supposons que $var_array est un tableau retourné
// par wddx_deserialize()
$taille = "grand";
$var_array = array("couleur" => "bleu",
                  "taille"  => "moyen",
                  "forme"   => "sphere");
extract($var_array, EXTR_PREFIX_SAME, "wddx");
echo "$couleur, $taille, $forme, $wddx_taille\n";
?>
```

L'exemple ci-dessus va afficher :

```
bleu, grand, sphere, moyen
```

La variable `$taille` n'a pas été réécrite, car on avait spécifié le paramètre `EXTR_PREFIX_SAME`, qui a permis la création de `$wddx_taille`. Si `EXTR_SKIP` avait été utilisée, alors `$wddx_taille` n'aurait pas été créé. Avec `EXTR_OVERWRITE`, `$taille` aurait pris la valeur "moyen", et avec `EXTR_PREFIX_ALL`, les variables créées seraient `$wddx_couleur`, `$wddx_taille` et `$wddx_forme`.

Vous devez utiliser un tableau associatif. Un tableau indexé numériquement ne produira aucun résultat, à moins que vous n'utilisiez l'option `EXTR_PREFIX_ALL` ou `EXTR_PREFIX_INVALID`.

Voir aussi `compact`.

8.3.63 in_array() : Indique si une valeur appartient à un tableau

bool **in_array** (mixed *needle* , array *haystack* , bool *strict*)

in_array recherche *needle* dans *haystack* et retourne TRUE s'il s'y trouve, ou FALSE sinon.

Le troisième paramètre *strict* est optionnel. S'il vaut TRUE alors in_array vérifiera aussi que le types du paramètre *needle* correspond au type de la valeur trouvée dans *haystack* .

Note
Si <i>needle</i> est une chaîne, la comparaison est faite en tenant compte de la casse.
Note
Dans les versions antérieure à 4.2.0, <i>needle</i> ne pouvait pas être un tableau.

Exemple avec <u>in_array</u>
<pre><?php \$os = array ("Mac", "NT", "Irix", "Linux"); if (in_array ("Linux", \$os)) { print "Trouvé Linux"; } if (in_array ("mac", \$os)) { print "Trouvé mac"; } ?></pre>
La seconde condition échoue, car <u>in_array</u> est sensible à la casse. Le script retourne :
Trouvé Linux

Exemple avec <u>in_array</u> et le mode strict
<pre><?php \$a = array('1.10', 12.4, 1.13); if (in_array('12.4', \$a, TRUE)) echo "'12.4' a été trouvé avec recherche stricte\n"; if (in_array(1.13, \$a, TRUE)) echo "1.13 a été trouvé avec recherche stricte\n"; ?></pre>
L'exemple ci-dessus va afficher :
1.13 a été trouvé avec recherche stricte

Exemple avec <u>in_array</u> et deux tableaux en paramètres
<pre><?php \$a = array(array('p', 'h'), array('p', 'r'), 'o'); if (in_array(array ('p', 'h'), \$a)) echo "'ph' a été trouvé\n"; if (in_array(array ('f', 'i'), \$a)) echo "'fi' a été trouvé\n"; if (in_array('o', \$a)) echo "'o' a été trouvé\n";</pre>

```
?>
```

L'exemple ci-dessus va afficher :

```
'ph' a été trouvé
```

```
'o' a été trouvé
```

Voir aussi [array_search](#) , [array_key_exists](#) et [isset](#) .

8.3.64 key() : Retourne une clé d'un tableau associatif

mixed **key** (array array)

key retourne la clé courante dans le tableau array .

Exemple avec [key](#)

```
<?php
$array = array(
    'fruit1' => 'pomme',
    'fruit2' => 'orange',
    'fruit3' => 'raisin',
    'fruit4' => 'pomme',
    'fruit5' => 'pomme');

// Cette boucle affiche toutes les clés
// dont la valeur vaut 'pomme'

while ($fruit_name = current($array)) {
    if ($fruit_name == 'pomme') {
        echo key($array) . '<br />';
    }
    next($array);
}
?>
```

Voir aussi [current](#) et [next](#) .

8.3.65 krsort() : Trie un tableau en sens inverse et suivant les clés

bool **krsort** (array array , int sort_flags)

krsort trie le tableau array en ordre inverse et suivant les clés, en maintenant la correspondance entre les clés et les valeurs. Cette fonction est pratique pour les tableaux associatifs.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec [krsort](#)

```
<?php
$fruits = array("d"=>"papaye", "a"=>"orange", "b"=>"banane", "c"=>"ananas");
krsort($fruits);
foreach ($fruits as $key => $val) {
    echo "$key = $val\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
d = papaye
c = ananas
b = banane
a = orange
```

Vous pouvez modifier le comportement de cette fonction en utilisant le paramètre optionnel `sort_flags` . Pour plus de détails, voyez le manuel pour la fonction [sort](#) .

Voir aussi [asort](#) , [arsort](#) , [ksort](#) , [sort](#) , [natsort](#) et [rsort](#) .

8.3.66 ksort() : Trie un tableau suivant les clés

bool **ksort** (array array , int sort_flags)

[ksort](#) trie le tableau array suivant les clés, en maintenant la correspondance entre les clés et les valeurs. Cette fonction est pratique pour les tableaux associatifs.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec [ksort](#)

```
<?php
$fruits = array("d"=>"papaye", "a"=>"orange", "b"=>"banane", "c"=>"ananas");
ksort($fruits);
foreach ($fruits as $key => $val) {
    echo "$key = $val\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
a => orange
b => banane
c => ananas
d => papaye
```

Vous pouvez modifier le comportement de cette fonction en utilisant le paramètre optionnel `sort_flags` . Pour plus de détails, voyez le manuel pour la fonction [sort](#) .

Note

Le second paramètre a été ajouté en PHP 4.0.

Voir aussi [asort](#) , [arsort](#) , [krsort](#) , [uksort](#) , [sort](#) , [natsort](#) et [rsort](#) .

8.3.67 list() : Transforme une liste de variables en tableau

void **list** (mixed varname , mixed ...)

Tout comme [array](#) , [list](#) n'est pas une véritable fonction, mais un élément de langage, qui permet de rassembler les variables varname , ... sous forme de tableau, pour les assigner en une seule ligne.

Note

`list` fonctionne uniquement avec des tableaux à indexation numérique, et suppose que l'indexation commence à 0.

Exemple avec `list`

```
<?php
$info = array('café', 'noir', 'cafeine');

// Liste de toutes les variables
list($drink, $color, $power) = $info;
echo "$drink est $color et la $power le rend particulier.\n";

// Liste de certaines variables
list($drink, , $power) = $info;
echo "$drink contient de la $power.\n";

// Ou bien, n'utilisons que le troisième
list( , , $power) = $info;
echo "J'ai besoin de $power!\n";

?>
```

Exemple avec `list` et MySQL

```
<table>
  <tr>
    <th>Nom de l'employé</th>
    <th>Salaire</th>
  </tr>

<?php
$resultat = mysql_query ("SELECT id, nom, salaire FROM employes", $conn);
while (list ($id, $nom, $salaire) = mysql_fetch_row ($resultat)) {
    echo " <tr>\n".
        "   <td><a href=\"info.php?id=$id\">$nom</a></td>\n".
        "   <td>$salaire</td>\n".
        " </tr>\n";
}

?>
</table>
```

Attention

`list` assigne les valeurs en commençant par la valeur la plus à droite. Si vous utiliser ces variables, ce ne sera pas un problème. Mais si vous utilisez des tableaux, vous serez surpris de voir que `list` les affecte de droite à gauche.

Exemple avec `list` en tenant compte de l'ordre

```
<?php

$info = array('café', 'noir', 'cafeine');

list($a[0], $a[1], $a[2]) = $info;

var_dump($a);

?>
```

Cet exemple donne le résultat suivant (notez l'ordre d'affichage des éléments, en comparaison avec l'ordre dans lequel ils ont été écrits dans la syntaxe de `list`) :

```
array(3) {
[2]=>
string(8) "cafeine"
[1]=>
string(5) "noir"
[0]=>
string(6) "café"
}
```

Voir aussi `each` , `array` et `extract` .

8.3.68 `natcasesort()` : Trie un tableau avec l'algorithme à "ordre naturel" insensible à la casse

bool `natcasesort` (array array)

`natcasesort` implémente un algorithme de tri qui traite les chaînes alpha-numériques du tableau `array` comme un être humain tout en conservant la relation clé/valeur. C'est ce qui est appelé l'"ordre naturel".

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

`natcasesort` est la version insensible à la casse de `natsort` .

Exemple avec `natcasesort`

```
<?php
$array1 = $array2 = array('IMG0.png', 'img12.png', 'img10.png',
'img2.png', 'img1.png', 'IMG3.png');

sort($array1);
echo "Standard sorting\n";
print_r($array1);

natcasesort($array2);
echo "\nNatural order sorting (case-insensitive)\n";
print_r($array2);
?>
```

L'exemple ci-dessus va afficher :

```
Standard sorting
Array
(
[0] => IMG0.png
[1] => IMG3.png
[2] => img1.png
[3] => img10.png
[4] => img12.png
[5] => img2.png
)
```

Natural order sorting (case-insensitive)

Array

```
(
[0] => IMG0.png
[4] => img1.png
[3] => img2.png
[5] => IMG3.png
[2] => img10.png
[1] => img12.png
)
```

Pour plus de détails, rendez-vous sur le site de Martin Pool : [Natural Order String Comparison](#)

Voir aussi [sort](#) , [natsort](#) , [strnatcmp](#) et [strnatcasecmp](#) .

8.3.69 natsort() : Trie un tableau avec l'algorithme à "ordre naturel"

bool **natsort** (array array)

[natsort](#) implémente un algorithme de tri qui traite les chaînes alpha-numériques du tableau array comme un être humain tout en conservant la relation clé/valeur. C'est ce qui est appelé l'"ordre naturel". Un exemple de la différence de traitement entre un tel algorithme et un algorithme de tri de chaînes (comme lorsqu'on utilise [sort](#)) est illustré ci-dessous :

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec [natsort](#)

```
<?php
$array1 = $array2 = array ("img12.png", "img10.png", "img2.png", "img1.png");

sort($array1);
echo "Tri standard\n";
print_r($array1);

natsort($array2);
echo "Tri en ordre naturel\n";
print_r($array2);
?>
```

L'exemple ci-dessus va afficher :

Tri standard

Array

```
(
[0] => img1.png
[1] => img10.png
[2] => img12.png
[3] => img2.png
)
```

Tri en ordre naturel

Array

```
(
[3] => img1.png
```

```
[2] => img2.png
[1] => img10.png
[0] => img12.png
)
```

Pour plus de détails, rendez-vous sur le site de Martin Pool : [Natural Order String Comparison](#) .
Voir aussi [natcasesort](#) , [strnatcmp](#) et [strnatcasecmp](#) .

8.3.70 next() : Avance le pointeur interne d'un tableau

mixed **next** (array array)

next retourne l'élément suivant du tableau array ou FALSE s'il n'y a plus d'éléments.

next se comporte comme current , mais avec une différence : il avance le pointeur interne de tableau d'un élément avant de retourner la valeur qu'il pointe. Lorsque le pointeur dépasse le dernier élément, next retourne FALSE .

Attention

Si le tableau contient des éléments vides ou des zéros, next retournera FALSE pour ces éléments. Pour passer proprement en revue un tableau de ce genre, il faut utiliser each .

Exemple avec next et ses amies

```
<?php
$transport = array('pied', 'velo', 'voiture', 'avion');
$mode = current($transport); // $mode = 'pied';
$mode = next($transport);    // $mode = 'velo';
$mode = next($transport);    // $mode = 'voiture';
$mode = prev($transport);    // $mode = 'velo';
$mode = end($transport);     // $mode = 'avion';
?>
```

Voir aussi [current](#) , [end](#) , [prev](#) et [reset](#) .

8.3.71 pos() : Alias de current

pos est un alias de la fonction current .

8.3.72 prev() : Recule le pointeur courant de tableau

mixed **prev** (array array)

prev repositionne le pointeur interne du tableau array à la dernière place qu'il occupait et en retourne la valeur, ou bien elle retourne FALSE s'il ne reste plus d'éléments.

Attention

Si le tableau contient des éléments vides, prev retournera FALSE pour ces éléments aussi. Pour passer en revue tous les éléments, utilisez plutôt each .

prev se comporte exactement comme next , mais elle fait reculer le pointeur plutôt que de l'avancer.

Exemple avec prev et ses amies

```
<?php
$transport = array('pied', 'velo', 'voiture', 'avion');
$mode = current($transport); // $mode = 'pied';
$mode = next($transport);    // $mode = 'velo';
$mode = next($transport);    // $mode = 'voiture';
$mode = prev($transport);    // $mode = 'velo';
$mode = end($transport);     // $mode = 'avion';
?>
```

Voir aussi [current](#) , [end](#) , [next](#) et [reset](#) .

8.3.73 range() : Crée un tableau contenant un intervalle d'éléments

array **range** (mixed low , mixed high , *number step*)

[range](#) retourne un tableau contenant tous les entiers depuis low jusqu'à high , inclus. Si low est supérieur à high , la séquence sera en ordre décroissant.

Note

Nouveau paramètre

Le paramètre optionnel step a été ajouté en PHP 5.0.

Si une valeur est donnée au paramètre step , il sera utilisé comme valeur incrémentale entre les éléments de la séquence. step doit être exprimé comme un nombre entier positif. S'il n'est pas spécifié, step vaut par défaut 1.

Exemple avec [range](#)

```
<?php
// array(0,1,2,3,4,5,6,7,8,9)
foreach(range(0, 9) as $number) {
    echo $number;
}

// La paramètre de pas (step) a été introduit en version 4.4.0
// array(0,10,20,30,40,50,60,70,80,90,100)
foreach(range(0, 100, 10) as $number) {
    echo $number;
}

// L'utilisation des caractères a été introduit en version 4.1.0
// array('a','b','c','d','e','f','g','h','i');
foreach(range('a', 'i') as $letter) {
    echo $letter;
}
// array('c','b','a');
foreach(range('c', 'a') as $letter) {
    echo $letter;
}
?>
```

Note

Avant la version PHP 4.1.0, la fonction [range](#) ne générait que des séquences d'entiers. Le support des séquences de caractères a été ajouté en PHP 4.1.0.

Attention

Dans les versions de PHP de 4.1.0 à 4.3.2, [range](#) voyait les chaînes numériques comme des chaînes et non comme des entiers. Au lieu de cela, ils étaient utilisés comme séquence de caractères. Par exemple, "4242" était traité comme "4" .

Voir aussi [shuffle](#) , [array_fill](#) et [foreach](#) .

8.3.74 reset() : Remet le pointeur interne de tableau au début

mixed **reset** (array array)

reset remplace le pointeur de tableau array au premier élément et retourne la valeur du premier élément ou FALSE si le tableau est vide.

Exemple avec [reset](#)

```
<?php
$array = array('1 kilomètre', '2 kilomètres', '3 kilomètres', '4 kilomètres');

// Par défaut, le pointeur est sur le premier élément
echo current($array) . "<br />\n"; // "1 kilomètre"

// on saute deux kilomètres
next($array);
next($array);
echo current($array) . "<br />\n"; // "3 kilomètres"

// on remet le pointeur au début
reset($array);
echo current($array) . "<br />\n"; // "1 kilomètre"
?>
```

Voir aussi [current](#) , [each](#) , [end](#) , [next](#) et [prev](#) .

8.3.75 rsort() : Trie un tableau en ordre inverse

bool **rsort** (array array , int sort_flags)

rsort effectue un tri en ordre décroissant (du plus grand au plus petit) du tableau array .

Note

Cette fonction assigne de nouvelles clés pour les éléments du paramètre array . Elle effacera toutes les clés existantes que vous aviez pu assigner, plutôt que de réordonner les clés.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec [rsort](#)

```
<?php
$fruits = array("papaye", "orange", "banane", "ananas");
rsort($fruits);
foreach ($fruits as $key => $val) {
    echo "$key = $val\n";
}
?>
```

L'exemple ci-dessus va afficher :

0 = papaye

```
1 = orange
2 = banane
3 = ananas
```

Les fruits ont été classés dans l'ordre alphabétique inverse.

Vous pouvez modifier le comportement de cette fonction en utilisant le paramètre optionnel `sort_flags` . Pour plus de détails, voyez le manuel pour la fonction [sort](#) .

Voir aussi [arsort](#) , [asort](#) , [ksort](#) , [krsort](#) , [sort](#) et [usort](#) .

8.3.76 shuffle() : Mélange les éléments d'un tableau

bool **shuffle** (array array)

[shuffle](#) mélange les éléments du tableau array .

Note

Cette fonction assigne de nouvelles clés pour les éléments du paramètre array . Elle effacera toutes les clés existantes que vous aviez pu assigner, plutôt que de réordonner les clés.

Exemple avec [shuffle](#)

```
<?php
$numbers = range(1,20);
srand((float)microtime()*1000000);
shuffle($numbers);
foreach ($numbers as $number) {
    echo "$number ";
}
?>
```

Note

Depuis PHP 4.2.0, vous n'avez plus besoin d'initialiser le générateur de nombres aléatoires avec [srand](#) ou [mt_srand](#) car c'est fait automatiquement.

Voir aussi [arsort](#) , [asort](#) , [ksort](#) , [rsort](#) , [sort](#) et [usort](#) .

8.3.77 sizeof() : Alias de [count](#)

[sizeof](#) est un [alias](#) de la fonction [count](#) .

8.3.78 sort() : Trie un tableau

bool **sort** (array array , int sort_flags)

[sort](#) trie le tableau array . Les éléments seront triés du plus petit au plus grand.

Note

Cette fonction assigne de nouvelles clés pour les éléments du paramètre array . Elle effacera toutes les clés existantes que vous aviez pu assigner, plutôt que de réordonner les clés.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec sort

```
<?php
$fruits = array("papaye", "orange", "banane", "ananas");
sort ($fruits);
foreach ($fruits as $key => $val) {
    echo "fruits[\".$key.\"] = ".$val."\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
fruits[0] = ananas
fruits[1] = banane
fruits[2] = orange
fruits[3] = papaye
```

Les fruits ont été classés dans l'ordre alphabétique.

Le paramètre optionnel `sort_flags` peut être utilisé pour modifier le comportement de trie en utilisant ces valeurs :

Constantes de type de tri :

- `SORT_REGULAR` : compare les éléments normalement (ne modifie pas les types)
- `SORT_NUMERIC` : compare les éléments numériquement
- `SORT_STRING` : compare les éléments comme des chaînes de caractères
- `SORT_LOCALE_STRING` : compare les éléments en utilisant la configuration locale. Ajouté en PHP 5.0.2 et 4.4.0. Avant PHP 6, il utilise les locales système, qui peuvent être modifiées en utilisant la fonction [setlocale](#) . Depuis PHP 6, vous devez utiliser la fonction [i18n_loc_set_default](#) .

Note

Le second paramètre a été ajouté en PHP 4.

Attention

Attention lorsque vous triez des tableaux avec des types différents de valeurs car le résultat de [sort](#) est imprévisible.

Voir aussi [arsort](#) , [asort](#) , [ksort](#) , [krsort](#) , [natsort](#) , [natcasesort](#) , [rsort](#) , [usort](#) , [array_multisort](#) et [uksort](#) .

8.3.79 uasort() : Trie un tableau en utilisant une fonction de callback

bool **uasort** (array array , callback cmp_function)

[uasort](#) trie le tableau `array` en conservant la correspondance entre les index et leurs valeurs. [uasort](#) sert essentiellement lors de tri de tableaux associatifs où l'ordre des éléments est significatif. La fonction de comparaison utilisée `cmp_function` est définie par l'utilisateur.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec uasort

```
<?php
function compare ($a, $b) {
```

```

    if ($a == $b) return 0;
    return ($a > $b) ? -1 : 1;
}

$a = array (4 => "quatre", 3 => "trois", 20 => "vingt", 10 => "dix");
uasort ($a, "compare");

while (list ($key, $value) = each ($a)) {
    echo "$key: $value\n";
}
?>

```

L'exemple ci-dessus va afficher :

```

20: vingt
3: trois
4: quatre
10: dix

```

Note

Voyez les fonctions [usort](#) et [uksort](#) pour des exemples de tris avec utilisation de fonction personnalisée.

Voir aussi [usort](#) , [uksort](#) , [sort](#) , [asort](#) , [arsort](#) , [ksort](#) et [rsort](#) .

8.3.80 [uksort\(\)](#) : Trie un tableau par ses clés en utilisant une fonction de callback

bool **uksort** (array array , callback cmp_function)

[uksort](#) trie les clés du tableau array en utilisant la fonction utilisateur cmp_function . Si un tableau doit être trié avec un critère complexe, il est préférable d'utiliser [uksort](#) .

La fonction cmp_function doit accepter deux paramètres, qui représenteront une paire de clé du tableau array . La fonction de comparaison doit retourner un entier supérieur, égal ou inférieur à zéro, pour, respectivement, indiquer que le premier argument est supérieur, égal ou inférieur au second.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec [uksort](#)

```

<?php
function compare ($a, $b) {
    if ($a == $b) return 0;
    return ($a > $b) ? -1 : 1;
}

$a = array (4 => "quatre", 3 => "trois", 20 => "vingt", 10 => "dix");

uksort ($a, "compare");

foreach ($a as $key => $value) {
    echo "$key: $value\n";
}
?>

```

L'exemple ci-dessus va afficher :

```
20: vingt
10: dix
4: quatre
3: trois
```

Voir aussi [usort](#) , [uasort](#) , [sort](#) , [asort](#) , [arsort](#) , [ksort](#) , [natsort](#) et [rsort](#) .

8.3.81 usort() : Trie un tableau en utilisant une fonction de comparaison

bool **usort** (array array , callback cmp_function)

[usort](#) va trier le tableau array avec ses valeurs, en utilisant une fonction définie par l'utilisateur. Si un tableau doit être trié avec un critère complexe, il est préférable d'utiliser cette méthode.

La fonction de comparaison cmp_function doit retourner un entier, qui sera inférieur, égal ou supérieur à zéro suivant que le premier argument est considéré comme plus petit, égal ou plus grand que le second argument. Si les deux arguments sont égaux, leur ordre est indéfini.

Note

Si deux éléments sont égaux, au sens de la fonction cmp_function , leur ordre sera indéfini. Jusqu'en PHP 4.0.6, la fonction de tri leur conserverait leur ordre original, mais le nouvel algorithme introduit en PHP 4.1.0 fait que ce n'est plus le cas, car cela serait trop coûteux.

Note

Cette fonction assigne de nouvelles clés pour les éléments du paramètre array . Elle effacera toutes les clés existantes que vous aviez pu assigner, plutôt que de réordonner les clés.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec [usort](#)

```
<?php
function compare($a, $b)
{
    if ($a == $b) {
        return 0;
    }
    return ($a > $b) ? -1 : 1;
}

$a = array (3, 2, 5, 6, 1);

usort ($a, "compare");

foreach ($a as $key => $value) {
    echo "$key: $value\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
0: 6
1: 5
2: 3
```

3: 2

4: 1

Note

Evidemment dans ce cas trivial, rsort serait plus approprié.

Tri avec usort sur un tableau multi-dimensionnel

```
<?php
function compare($a, $b)
{
    return strcmp($a["fruit"], $b["fruit"]);
}

$fruits[0]["fruit"] = "citron";
$fruits[1]["fruit"] = "pommes";
$fruits[2]["fruit"] = "raisins";

usort($fruits, "compare");

while (list ($key, $value) = each ($fruits)) {
    echo "\$fruits[$key]: " . $value["fruit"] . "\n";
}
?>
```

Lors du tri de tableau multi-dimensionnel, \$a et \$b contiennent des références sur le premier élément du tableau.

L'exemple ci-dessus va afficher :

```
$fruits[0]: raisins
$fruits[1]: pommes
$fruits[2]: citron
```

Tri avec usort sur un objet

```
<?php
class TestObj {
    var $name;

    function TestObj($name)
    {
        $this->name = $name;
    }

    /* Ceci est une fonction de comparaison statique */
    function cmp_obj($a, $b)
    {
        $a1 = strtolower($a->name);
        $b1 = strtolower($b->name);
        if ($a1 == $b1) {
            return 0;
        }
        return ($a1 > $b1) ? +1 : -1;
    }
}

$a[] = new TestObj("c");
$a[] = new TestObj("b");
$a[] = new TestObj("d");

uasort($a, array ("TestObj", "cmp_obj"));
```

```
foreach ($a as $item) {  
    echo $item->name."\n";  
}  
?>
```

L'exemple ci-dessus va afficher :

```
b  
c  
d
```

Voir aussi [uasort](#) , [uksort](#) , [sort](#) , [asort](#) , [arsort](#) , [ksort](#) , [natsort](#) et [rsort](#) .

8.4 Aspell [Obs.]

8.4.1 Introduction

Les fonctions Aspell vous permettent de vérifier l'orthographe d'un mot, et d'offrir des suggestions de correction.

Note

Cette extension a été retirée de PHP et n'est plus disponible depuis PHP 4.3.0. Si vous voulez utiliser un outil de correction orthographique avec PHP, voyez plutôt [pspell](#). Elle utilise la bibliothèque pspell et fonctionne avec les nouvelles versions de Aspell.

8.4.2 Pré-requis

Aspell fonctionne avec de très vieilles versions (jusqu'à la version .27.* ou presque) de la bibliothèque Aspell. Ce module, et ces versions d'Aspell ne sont plus maintenues ou supportées. Si vous en avez besoin, la librairie Aspell est disponible sur <http://aspell.sourceforge.net/>.

8.4.3 Installation

En PHP 4, ces fonctions ne sont disponibles que si PHP a été compilé avec l'option `--with-aspell[=DIR]`.

8.4.4 Voir aussi

Voir aussi [pspell](#).

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Voir aussi](#)
- [aspell_check_raw](#)
- [aspell_check](#)
- [aspell_new](#)
- [aspell_suggest](#)

8.4.6 `aspell_check()` : Vérifie un mot [Obsolète]

`bool aspell_check (int dictionary_link , string word)`

`aspell_check` vérifie l'orthographe d'un mot et retourne TRUE si l'orthographe est correcte, FALSE sinon.

Exemple avec `aspell_check`

```
<?php
$aspell_link = aspell_new("french");
if (aspell_check($aspell_link, "testt")) {
```

```

    echo "L'orthographe est correcte.";
} else {
    echo "Désolé, l'orthographe est incorrecte.";
}
?>

```

8.4.7 `aspell_new()` : Charge un nouveau dictionnaire [Obsolète]

int **aspell_new** (string *master* , string *personal*)

`aspell_new` ouvre un nouveau dictionnaire et retourne un identifiant de dictionnaire pour utilisation ultérieure avec les fonctions `aspell`. `aspell_new` retourne FALSE en cas d'erreur.

Exemple avec `aspell_new`

```

<?php
$aspell_link = aspell_new("french");
?>

```

8.4.8 `aspell_suggest()` : Suggère l'orthographe d'un mot [Obsolète]

array **aspell_suggest** (int *dictionary_link* , string *word*)

`aspell_suggest` retourne un tableau contenant les orthographes possibles d'un mot mal orthographié.

Exemple avec `aspell_suggest`

```

<?php
$aspell_link = aspell_new("french");

if (!aspell_check($aspell_link, "testt")) {
    $suggestions = aspell_suggest($aspell_link, "testt");

    foreach ($suggestions as $suggestion) {
        echo "Orthographe envisageable : $suggestion<br />\n";
    }
}
?>

```

8.5 Nombres de grande taille BCMath

8.5.1 Introduction

Cette extension permet de réaliser des calculs sur des nombres de précision arbitraire : la Calculatrice Binaire (Binary Calculator) supporte des nombres de toutes taille et précision, représentés sous forme de chaînes.

8.5.2 Pré-requis

Depuis PHP 4.0.4, libbcmath est distribuée avec PHP. Vous n'avez besoin d'aucune bibliothèque externe pour cette extension.

8.5.3 Installation

Ces fonctions ne sont disponibles que si l'option de configuration `--enable-bcmath` a été activée lors de la compilation. En PHP 3, ces fonctions ne sont disponibles que si l'option de configuration `--disable-bcmath` n'a pas été activée lors de la compilation.

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de bibliothèque supplémentaire pour disposer de ces fonctions.

8.5.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable	Historique
<code>bcmath.scale</code>	"0"	PHP_INI_ALL	Disponible depuis PHP 5.0.0.

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

`bcmath.scale` [entier](#)

Le nombre de décimales pour toutes les fonctions `bcmath`. Voir aussi [bcscale](#).

8.5.5 Types de ressources

Cette extension ne définit aucune ressource.

8.5.6 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)

- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [bcadd](#)
- [bccomp](#)
- [bcdiv](#)
- [bcmmod](#)
- [bcmul](#)
- [bcpow](#)
- [bcpowmod](#)
- [bcscale](#)
- [bcsqrt](#)
- [bcsub](#)

8.5.8 **bccomp()** : Compare deux nombres de grande taille

int **bccomp** (string left_operand , string right_operand , int scale)

[bccomp](#) compare l'opérande left_operand avec l'opérande right_operand et renvoie le résultat sous forme de valeur numérique (entier). Le paramètre optionnel scale est utilisé pour définir le nombre de chiffres après la virgule utilisés lors de la comparaison. Le résultat est 0 si les deux opérandes sont égaux. Si l'opérande left_operand est plus grand que l'opérande right_operand , le résultat est 1. Si l'opérande left_operand est plus petit que l'opérande right_operand , le résultat est -1.

8.5.9 **bcdiv()** : Divise deux nombres de grande taille

string **bcdiv** (string left_operand , string right_operand , int scale)

[bcdiv](#) divise l'opérande left_operand par l'opérande right_operand et renvoie le résultat. Le paramètre optionnel scale définit le nombre de chiffres après la virgule dans le résultat et par défaut vaut 0.

8.5.10 **bcmmod()** : Retourne le reste d'une division entre nombres de grande taille

string **bcmmod** (string left_operand , string modulus)

[bcmmod](#) retourne le reste de la division entre left_operand en utilisant modulus .

8.5.11 **bcmul()** : Multiplie deux nombres de grande taille

string **bcmul** (string left_operand , string right_operand , int scale)

[bcmul](#) multiplie l'opérande left_operand par l'opérande right_operand et renvoie le résultat. Le paramètre optionnel scale définit le nombre de chiffres après la virgule dans le résultat.

8.5.12 bcpow() : Elève un nombre à la puissance n-ième

string **bcpow** (string *x* , string *y* , int *scale*)

bcpow élève *x* à la puissance *y* . Le paramètre optionnel *scale* définit le nombre de chiffres après la virgule dans le résultat.

8.5.13 bcpowmod() : Calcule le reste modulo d'un nombre élevé à une puissance

string **bcpowmod** (string *x* , string *y* , string *modulus* , int *scale*)

bcpowmod utilise la méthode d'exponentiation rapide pour élever le nombre *x* à la puissance *y* , et en calculant le reste modulo *modulus* . Le paramètre optionnel *scale* sert à indiquer le nombre de décimales à utiliser dans le résultat.

Note

Comme cette méthode utilise les opérations de modulo, les nombres non naturels risquent de donner des résultats inattendus. Un nombre naturel est un entier positif non-nul.

8.5.14 bcscale() : Détermine le nombre de décimales par défaut

bool **bcscale** (int *scale*)

bcscale définit la précision par défaut pour toutes les fonctions mathématiques sur des nombres de taille arbitraire qui suivent et qui omettent le paramètre *scale* . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

8.5.15 bcsqrt() : Renvoie la racine carrée d'un nombre de grande taille

string **bcsqrt** (string *operand* , int *scale*)

bcsqrt renvoie la racine carrée de l'opérande *operand* . Le paramètre optionnel *scale* définit le nombre de chiffres après la virgule dans le résultat.

8.5.16 bcsub() : Soustrait un nombre de grande taille à un autre

string **bcsub** (string *left_operand* , string *right_operand* , int *scale*)

bcsub soustrait l'opérande *right_operand* à l'opérande *left_operand* et renvoie le résultat sous forme de chaîne de caractères. Le paramètre optionnel *scale* définit le nombre de chiffres après la virgule dans le résultat.

8.6 Compilateur bytecode pour PHP

8.6.1 Introduction

Attention

Ce module est **EXPERIMENTAL** . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, **SANS PREAVIS!** Soyez-en conscient, et utilisez ce module à vos risques et périls.

Bcompiler a été écrit pour plusieurs raisons :

- Pour encoder un script complet dans une application PHP propriétaire
- Pour encoder des classes et/ou des fonctions dans une application PHP propriétaire
- Pour permettre d'utiliser des applications php-gtk sur des bureaux clients sans avoir besoin du fichier php.exe.
- Pour rendre faisable de convertir un code PHP en C

Le premier de ces buts est réalisé en utilisant les fonctions bcompiler_write_header , bcompiler_write_file et bcompiler_write_footer . Les fichiers de bytecode peuvent être écrits en non-compressés ou en texte plein. Pour utiliser le bytecode généré, vous pouvez simplement l'inclure avec les structures include ou require.

Le second but est atteint en utilisant les fonctions bcompiler_write_header , bcompiler_write_class , bcompiler_write_footer , bcompiler_read et bcompiler_load . Les fichiers bytecode peuvent être écrits en non-compressés ou en texte plein. La fonction bcompiler_load lit un fichier bytecode compressé bzipé, qui tend à avoir un taille d'1/3 par rapport au fichier original.

Pour créer des fichiers du type EXE, bcompiler doit être utilisé avec un fichier sapi modifié ou une version de PHP qui a été compilé en tant que bibliothèque partagée. Dans ce cas, bcompiler lit le bytecode compressé depuis la fin du fichier exe.

bcompiler peut augmenter d'environ 30% ces performances lorsqu'il est utilisé avec uniquement du bytecode non-compressé. Mais gardé en mémoire que le bytecode non-compressé peut être 5 fois plus long (voir plus) que le code source original. Utiliser la compression bytecode peut vous permettre de réduire l'espace disque mais la décompression requiert plus de temps que de parser simplement le code source. bcompiler n'effectue aucune optimisation du bytecode, cette fonctionnalité pourra être ajoutée dans le futur...

bcompiler n'est pas susceptible d'améliorer les performances d'exécutions, car il ne fait que dévier l'étape de compilation du processus courant PHP et la remplace par le processus d'importation dans bcompiler. Il n'optimise pas le bytecode, ce qui pourrait être ajouté dans le futur.

En terme de sécurité du code, il est possible de dire qu'il est impossible de recréer le code source exact depuis lequel il a été construit et sans les commentaires accompagnant le code source. Il serait inefficace d'utiliser bcompiler pour recréer et modifier une classe. Malgré tout, il reste possible de récupérer les données depuis un fichier bytecode encodé avec bcompiler, donc, n'y mettez pas vos mots de passe ou toute autre donnée de ce genre.

8.6.2 Installation

Note succincte d'installation :

- Vous avez besoin d'au moins PHP 4.3 pour que la compression fonctionne
- Pour l'installer sur PHP 4.3 et suivant, à l'invite de commande Unix, tapez la commande :
pear install bcompiler
- Pour l'installer sous Windows, tant que le mécanisme de distribution de paquets n'est pas terminé, merci de rechercher dans les archives de la liste de diffusion pear-general les paquets pré-construits. (ou envoyer un email sur cette liste si vous ne trouvez aucune référence)
- Pour l'installer sur une version plus ancienne, vous devez effectuer quelques modifications à la construction.
- décompressez l'archive bcompiler.tgz dans le dossier php4/ext . (récupérez le directement depuis PECL <http://pecl.php.net/get/bcompiler>)
- Si le nouveau dossier porte un nom comme bcompiler-0.x , alors, renommez le en bcompiler (mise à part si vous voulez le construire comme module php autonome).
- Si vous utilisez une version plus ancienne que PHP 4.3, vous devez copier le fichier Makefile.in.old dans Makefile.in et le fichier config.m4.old dans config.m4 .
- exécutez la commande phpize dans le dossier ext/bcompiler
- exécutez la commande ./buildconf dans le dossier php4
- exécutez la commande configure avec l'option --enable-bcompiler (ainsi que vos autres options)
- make; make install
- that's it.

8.6.3 Contact

Si vous avez des commentaires, des corrections de bogues, des améliorations ou que vous voulez aider à rendre cette extension meilleure, vous pouvez envoyer un email à alan_k@php.net . Toute aide est vraiment la bienvenue.

- [Introduction](#)
- [Installation](#)
- [Contact](#)
- [bcompiler_load_exe](#)
- [bcompiler_load](#)
- [bcompiler_parse_class](#)
- [bcompiler_read](#)
- [bcompiler_write_class](#)
- [bcompiler_write_constant](#)
- [bcompiler_write_exe_footer](#)
- [bcompiler_write_file](#)
- [bcompiler_write_footer](#)
- [bcompiler_write_function](#)
- [bcompiler_write_functions_from_file](#)
- [bcompiler_write_header](#)

8.6.5 bcompiler_load() : Lit et crée les classes depuis un fichier compressé bzippé

bool **bcompiler_load** (string filename)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Note

Utiliser les structures include ou require pour parser les bytecodes, c'est plus portable et plus commode que d'utiliser cette fonction.

`bcompiler_load` lit les données depuis le fichier compressé bzippé filename et crée les classes depuis le bytecode. Notez que cette fonction n'exécute pas le corps du script contenu dans le fichier bytecode.

Exemple avec `bcompiler_load`

```
<?php
bcompiler_load("/tmp/example");
print_r(get_defined_classes());
?>
```

8.6.6 `bcompiler_parse_class()` : Lit le bytecode d'une classe et revient à une fonction utilisateur

bool `bcompiler_parse_class` (string class , string callback)

Note

Cette fonction a été supprimée de `bcompiler` et n'est plus disponible depuis `bcompiler 0.5`.

`bcompiler_parse_class` lit le bytecode d'une classe et revient à une fonction utilisateur.

Exemple avec `bcompiler_parse_class`

```
<?php
function readByteCodes($data) {
    print_r($data);
}
bcompiler_parse_class("DB", "readByteCodes");
?>
```

8.6.7 `bcompiler_read()` : Lit et crée les classes depuis un descripteur de fichier

bool `bcompiler_read` (resource filehandle)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Note

Utiliser les structures include ou require pour parser les bytecodes, c'est plus portable et plus commode que d'utiliser cette fonction.

`bcompiler_read` lit les données depuis un fichier ouvert représenté par le descripteur filehandle et crée les classes depuis le bytecode. Notez que cette fonction n'exécute pas le corps du script contenu dans le fichier bytecode.

Exemple avec `bcompiler_read`

```
<?php
$fh = fopen("/tmp/example", "r");
bcompiler_read($fh);
fclose($fh);
print_r(get_defined_classes());

?>
```

8.6.8 `bcompiler_write_class()` : Ecrit une classe définie en bytecode

bool `bcompiler_write_class` (resource filehandle , string className , string extends)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`bcompiler_write_class` lit le bytecode d'une classe existante nommée className depuis PHP et l'écrit dans le fichier ouvert désigné par le descripteur filehandle . Cela n'effectue pas de vérification sur les dépendances, donc, assurez-vous d'écrire les classes dans l'ordre pour éviter d'avoir une alerte du genre ' undefined class ' lorsque vous les chargez.

Exemple avec `bcompiler_write_class`

```
<?php
$fh = fopen("/tmp/example", "w");
bcompiler_write_header($fh);
bcompiler_write_class($fh, "DB");
// you must write DB_common before DB_mysql, as DB_mysql extends DB_common.
bcompiler_write_class($fh, "DB_common");
bcompiler_write_class($fh, "DB_mysql");
bcompiler_write_footer($fh);
fclose($fh);

?>
```

Voir aussi `bcompiler_write_header` et `bcompiler_write_footer` .

8.6.9 `bcompiler_write_constant()` : Ecrit une constante définie comme bytecode

bool `bcompiler_write_constant` (resource filehandle , string constantName)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`bcompiler_write_constant` lit le bytecode depuis PHP pour une constante existante constantName et écrit le bytecode dans le fichier ouvert désigné par filehandle .

Exemple avec `bcompiler_write_constant`

```
<?php
define("MODULE_MAX", 30);

$fh = fopen("/tmp/example", "w");
bcompiler_write_header($fh);
bcompiler_write_constant($fh, "MODULE_MAX");
bcompiler_write_footer($fh);
fclose($fh);

?>
```

Voir aussi `bcompiler_write_header` et `bcompiler_write_footer` .

8.6.10 `bcompiler_write_exe_footer()` : Ecrit à la position `start` et jusqu'à la fin du fichier de type `exe`

bool `bcompiler_write_exe_footer` (resource filehandle , int startpos)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Un fichier `exe` (ou auto-exécutable) est composé de trois parties :

- Le stub (code exécutable, e.g. programme C compilé) qui charge l'interpréteur PHP, l'extension `bcompiler`, stocke le bytecode et initialise un appel à la fonction spécifique (e.g. `main`) ou une méthode d'une classe (i.e. `main::main`)
- Le bytecode (uniquement décompressé pour le moment)
- Le footer EXE `bcompiler`

Le paramètre `startpos` est la position à laquelle le bytecode commence et peut être obtenue en utilisant `ftell($fh)` .

Pour obtenir un stub approprié, vous pouvez compiler le stub `php_embed-based` `phpe.c` qui se trouve dans le dossier `exemples/embed` sur le CVS de `bcompiler`.

Exemple avec `bcompiler_write_footer`

```
<?php
/* Création du fichier de sortie (example.exe) */
$fh = fopen("example.exe", "w");
/* 1) Ecriture d'un stub (phpe.exe) */
$size = filesize("phpe.exe");
$fr = fopen("phpe.exe", "r");
fwrite($fh, fread($fr, $size), $size);
$startpos = ftell($fh);
/* 2) Ecriture du bytecode */
bcompiler_write_header($fh);
bcompiler_write_class($fh, "myclass");
bcompiler_write_function($fh, "main");
bcompiler_write_footer($fh);
/* 3) Ecriture du footer EXE */
bcompiler_write_exe_footer($fh, $startpos);
/* Fermeture du fichier de sortie */
fclose($fh);

?>
```

Voir aussi [bcompiler_write_header](#) , [bcompiler_write_class](#) et [bcompiler_write_footer](#) .

8.6.11 `bcompiler_write_file()` : Ecrit un code source PHP sous forme de bytecodes

bool `bcompiler_write_file` (resource filehandle , string filename)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`bcompiler_write_file` compile le fichier filename en bytecodes, et écrit le résultat dans le fichier filename .

Exemple avec `bcompiler_write_file`

```
<?php
$fh = fopen("example.phb", "w");
bcompiler_write_header($fh);
bcompiler_write_file($fh, "example.php");
bcompiler_write_footer($fh);
fclose($fh);
/* the following should be equivalent:
include "example.php";
    and
include "example.phb";
*/
?>
```

Voir aussi [bcompiler_write_header](#) et [bcompiler_write_footer](#) .

8.6.12 `bcompiler_write_footer()` : Ecrit le caractère \x00 pour indiquer la fin des données compilées

bool `bcompiler_write_footer` (resource filehandle)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`bcompiler_write_footer` écrit le caractère \x00 pour indiquer la fin des données compilées.

Exemple avec `bcompiler_write_footer`

```
<?php
$fh = fopen("/tmp/example", "w");
bcompiler_write_header($fh);
bcompiler_write_class($fh, "DB");
bcompiler_write_class($fh, "DB_common");
bcompiler_write_footer($fh);
fclose($fh);

?>
```

Voir aussi [bcompiler_write_header](#) et [bcompiler_write_header](#) .

8.6.13 `bcompiler_write_function()` : Écrit une fonction définie sous forme de bytecode

bool `bcompiler_write_function` (resource filehandle , string functionName)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`bcompiler_write_function` lit le bytecode d'une fonction existante depuis PHP et l'écrit dans le fichier désigné par la ressource de fichier filehandle . L'ordre n'est pas important (e.g. si la fonction b utilise la fonction a et que vous compilez l'exemple ci-dessous, cela fonctionnera très bien).

Exemple avec `bcompiler_write_function`

```
<?php
$fh = fopen("/tmp/example", "w");
bcompiler_write_header($fh);
bcompiler_write_function($fh, "my_function_a");
bcompiler_write_function($fh, "my_function_b");
bcompiler_write_footer($fh);
fclose($fh);

?>
```

Voir aussi `bcompiler_write_header` et `bcompiler_write_footer` .

8.6.14 `bcompiler_write_functions_from_file()` : Écrit toutes les fonctions définies dans un fichier sous forme de bytecode

bool `bcompiler_write_functions_from_file` (resource filehandle , string fileName)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`bcompiler_write_functions_from_file` recherche toutes les fonctions déclarées dans le fichier nommé fileName et écrit leurs bytecodes correspondants dans le fichier désigné par le descripteur filehandle . Souvenez-vous de toujours include/require les fichiers que vous tentez de compiler.

Exemple avec `bcompiler_write_functions_from_file`

```
<?php
require('module.php');

$fh = fopen("/tmp/example", "w");
bcompiler_write_header($fh);
bcompiler_write_functions_from_file($fh, 'module.php');
bcompiler_write_footer($fh);
fclose($fh);

?>
```

Voir aussi `bcompiler_write_header` et `bcompiler_write_footer` .

8.6.15 `bcompiler_write_header()` : Ecrit l'en-tête `bcompiler`

bool `bcompiler_write_header` (resource filehandle , string write_ver)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`bcompiler_write_header` écrit la partie d'en-tête d'un fichier `bcompiler`. Le paramètre optionnel `write_ver` peut être utilisé pour écrire le bytecode dans un format utilisé précédemment, vous pouvez donc l'utiliser avec les anciennes versions de `bcompiler`.

Exemple avec `bcompiler_write_header`

```
<?php
$fh = fopen("/tmp/example", "w");
bcompiler_write_header($fh);
bcompiler_write_class($fh, "DB");
bcompiler_write_footer($fh);
fclose($fh);

?>
```

Voir aussi [bcompiler_write_file](#) , [bcompiler_write_class](#) , [bcompiler_write_function](#) et [bcompiler_write_footer](#) .

8.7 Compression Bzip2

8.7.1 Introduction

Les fonctions bzip2 servent à lire et écrire dans les fichiers compressés avec la méthode bzip2 (.bz2).

8.7.2 Pré-requis

Ce module utilise les fonctions de la bibliothèque [bzip2](#), de Julian Seward pour écrire et lire des fichiers bzip2 (.bz2) de manière transparente. Ce module requiert une version de bzip2/libbzip2 >= 1.0.x.

8.7.3 Installation

Le support BZIP2 par PHP n'est pas activé par défaut. Vous devez utiliser l'option de configuration `--with-bz2[=DIR]` lors de la compilation de PHP pour l'activer.

8.7.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.7.5 Types de ressources

Cette extension définit un type de ressource : une ressource de fichier, identifiant un fichier compressé bzip2.

8.7.6 Constantes pré-définies

Cette extension ne définit aucune constante.

8.7.7 Exemples

Cet exemple ouvre un fichier temporaire, et écrit une ligne de test, puis il en affiche le contenu.

```
Petit exemple avec bzip2
<?php
$filename = "/tmp/fichier_de_test.bz2";
$str = "Ceci est une chaîne de test.\n";

// ouvre le fichier en écriture
$bz = bzopen($filename, "w");

// écrit une chaîne dans le fichier
bzwrite($bz, $str);

// ferme le fichier
```

```

bzclosé($bz);

// ouvre le fichier en lecture
$bz = bzopen($filename, "r");

// lit 10 caractères
echo bzread($bz, 10);

// affiche tout le reste du fichier, puis le ferme
echo bzread($bz);

bzclosé($bz);
?>

```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [bzclosé](#)
- [bzcompress](#)
- [bzdecompress](#)
- [bzerrno](#)
- [bzerror](#)
- [bzerrstr](#)
- [bzflush](#)
- [bzopen](#)
- [bzread](#)
- [bzwrite](#)

8.7.9 bzcompress() : Comprime une chaîne avec bzip2

mixed **bzcompress** (string source , int blocksize , int workfactor)

[bzcompress](#) comprime la chaîne source et retourne les données ainsi encodées.

8.7.10 bzdecompress() : Décomprime une chaîne bzip2

mixed **bzdecompress** (string source , int small)

[bzdecompress](#) décomprime la chaîne source , contenant des données compressées bzip2.

8.7.11 bzerrno() : Retourne le code d'erreur bzip2

int **bzerrno** (resource bz)

[bzerrno](#) retourne le code d'erreur du fichier bz2 représenté par le pointeur bz .

8.7.12 **bzerror()** : Retourne le numéro et le message d'erreur bzip2 dans un tableau

array **bzerror** (resource bz)

bzerror retourne un tableau associatif avec le numéro et le message d'erreur du fichier bz2 représenté par le pointeur bz .

8.7.13 **bzerrstr()** : Retourne le message d'erreur bzip2

string **bzerrstr** (resource bz)

bzerrstr retourne le message d'erreur du fichier bz2 représenté par le pointeur bz .

8.7.14 **bzflush()** : Force l'écriture de toutes les données compressées

int **bzflush** (resource bz)

bzflush force l'écriture de toutes les données bzip2 mises en tampon pour le fichier représenté par bz .

8.7.15 **bzopen()** : Ouvre un fichier compressé avec bzip2

resource **bzopen** (string filename , string mode)

bzopen ouvre un fichier bzip2 (.bz2) en écriture ou en lecture.

8.7.16 **bzread()** : Lecture binaire d'un fichier bzip2

string **bzread** (resource bz , int length)

bzread lit depuis le pointeur de fichier bzip2 donné.

La lecture s'arrête lorsque length (non-compressé) octets ont été lus ou si la fin du fichier est atteint, le premier des deux qui survient.

8.7.17 **bzwrite()** : Ecriture binaire dans un fichier bzip2

int **bzwrite** (resource bz , string data , int length)

bzwrite écrit le contenu de la chaîne data dans le fichier bzip2 représenté par bz .

8.8 Calendrier

8.8.1 Introduction

L'extension de calendrier propose une série de fonctions qui simplifient les conversions entre les différents formats de calendrier. La référence est le nombre de jours du calendrier Julien. C'est le nombre de jours depuis une date qui commence bien au-delà des dates les plus reculées dont on a besoin (située en 4000 avant J.C.). Pour convertir une date d'un calendrier à un autre, il faut d'abord la convertir dans ce calendrier, puis convertir le résultat dans le calendrier désiré. Attention, le nombre de jours du calendrier Julien est un système très différent du calendrier Julien! Pour plus d'informations, rendez-vous sur le site de <http://www.hermetic.ch/cal Stud/jdn.htm> . Pour plus d'informations (en anglais), reportez-vous à <http://www.boogle.com/info/cal-overview.html> . Les traductions issues de ces pages seront mises entre guillemets.

8.8.2 Installation

Pour faire fonctionner ces fonctions avec PHP, vous devez le compiler avec l'option `--enable-calendar` .

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de bibliothèque supplémentaire pour disposer de ces fonctions.

8.8.3 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.8.4 Types de ressources

Cette extension ne définit aucune ressource.

8.8.5 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

- CAL_GREGORIAN (entier)
- CAL_JULIAN (entier)
- CAL_JEWISH (entier)
- CAL_FRENCH (entier)
- CAL_NUM_CALS (entier)
- CAL_DOW_DAYNO (entier)
- CAL_DOW_SHORT (entier)
- CAL_DOW_LONG (entier)
- CAL_MONTH_GREGORIAN_SHORT (entier)
- CAL_MONTH_GREGORIAN_LONG (entier)
- CAL_MONTH_JULIAN_SHORT (entier)
- CAL_MONTH_JULIAN_LONG (entier)
- CAL_MONTH_JEWISH (entier)

CAL_MONTH_FRENCH ([entier](#))

Les constantes suivantes sont disponibles depuis PHP 4.3.0 :

CAL_EASTER_DEFAULT ([entier](#))

CAL_EASTER_ROMAN ([entier](#))

CAL_EASTER_ALWAYS_GREGORIAN ([entier](#))

CAL_EASTER_ALWAYS_JULIAN ([entier](#))

Les constantes suivantes sont disponibles depuis PHP 5.0.0 :

CAL_JEWISH_ADD_ALAFIM_GERESH ([entier](#))

CAL_JEWISH_ADD_ALAFIM ([entier](#))

CAL_JEWISH_ADD_GERESHAYIM ([entier](#))

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [cal_days_in_month](#)
- [cal_from_jd](#)
- [cal_info](#)
- [cal_to_jd](#)
- [easter_date](#)
- [easter_days](#)
- [FrenchToJD](#)
- [GregorianToJD](#)
- [JDDayOfWeek](#)
- [JDMonthName](#)
- [JDToFrench](#)
- [JDToGregorian](#)
- [jdtoweb](#)
- [JDToJulian](#)
- [jdtounix](#)
- [JewishToJD](#)
- [JulianToJD](#)
- [unixtojd](#)

8.8.7 cal_from_jd() : Convertit le nombre de jours Julien en un calendrier et retourne des détails

array [cal_from_jd](#) (int *jd* , int *calendar*)

[cal_from_jd](#) convertit le nombre de jours Julien *jd* en une date du calendrier *calendar* . Les valeurs possibles pour *calendar* sont CAL_GREGORIAN , CAL_JULIAN , CAL_JEWISH et CAL_FRENCH .

Exemple avec [cal_from_jd](#)

```
<?php
$today = unixtojd(mktime(0, 0, 0, 8, 16, 2003));
print_r(cal_from_jd($today, CAL_GREGORIAN));
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[date] => 8/16/2003
[month] => 8
[day] => 16
[year] => 2003
[dow] => 6
[abbrevdayname] => Sat
[dayname] => Saturday
[abbrevmonth] => Aug
[monthname] => August
)
```

Voir aussi [cal_to_jd](#) .

8.8.8 cal_info() : Retourne des détails sur un calendrier

array **cal_info** (*int* *calendar*)

[cal_info](#) retourne des informations sur le calendrier *calendar* spécifié.

Les informations de calendriers sont retournées dans un tableau qui contient les éléments *calname* , *calsymbol* , *month* , *abbrevmonth* et *maxdaysinmonth* . Les noms des différents calendriers qui peuvent être utilisés dans le paramètre *calendar* sont les suivants :

- 0 ou CAL_GREGORIAN - Calendrier Grégorien
- 1 ou CAL_JULIAN - Calendrier Julien
- 2 or CAL_JEWISH - Calendrier Jewish
- 3 or CAL_FRENCH - Calendrier révolutionnaire français

Si le paramètre *calendar* n'est pas spécifié, un tableau représentant tous les calendriers est retourné. Cette fonctionnalité est disponible à partir de PHP 5.

[cal_info](#) example

```
<?php
$info = cal_info(0);
print_r($info);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[months] => Array
(
[1] => January
[2] => February
[3] => March
[4] => April
```

```

[5] => May
[6] => June
[7] => July
[8] => August
[9] => September
[10] => October
[11] => November
[12] => December
)

[abbrevmonths] => Array
(
[1] => Jan
[2] => Feb
[3] => Mar
[4] => Apr
[5] => May
[6] => Jun
[7] => Jul
[8] => Aug
[9] => Sep
[10] => Oct
[11] => Nov
[12] => Dec
)

[maxdaysinmonth] => 31
[calname] => Gregorian
[calsymbol] => CAL_GREGORIAN
)

```

8.8.9 `cal_to_jd()` : Convertit un calendrier en nombre de jours Julien

int `cal_to_jd` (int `calendar` , int `month` , int `day` , int `year`)

`cal_to_jd` calcule le nombre de jours Juliens pour une date, dans le calendrier `calendar` . Les valeurs possibles pour `calendar` sont `CAL_GREGORIAN` , `CAL_JULIAN` , `CAL_JEWISH` et `CAL_FRENCH` .

Voir aussi `cal_from_jd` .

8.8.10 `easter_date()` : Retourne un timestamp UNIX pour Pâques, à minuit

int `easter_date` (int `year`)

`easter_date` retourne un timestamp UNIX pour Pâques, à minuit, pour une année donnée.

Depuis PHP 4.3.0, le paramètre `year` est optionnel, et prend la valeur de l'année en cours par défaut.

Exemple avec `easter_date`

```
<?php
echo date("M-d-Y", easter_date(1999));      /* "Apr-04-1999" */
echo date("M-d-Y", easter_date(2000));     /* "Apr-23-2000" */
echo date("M-d-Y", easter_date(2001));     /* "Apr-15-2001" */

?>
```

Attention

easter_date génère une alerte (Warning) si la date tombe hors de la zone de validité des timestamp UNIX (i.e. avant 1970 ou après 2037).

La date de Pâques a été fixée par le concile de Nicée, en 325 de notre ère, comme étant le dimanche après la première pleine lune qui suit l'équinoxe de printemps. L'équinoxe de printemps est considéré comme étant toujours le 21 mars, ce qui réduit le problème au calcul de la date de la lune pleine qui suit, et le dimanche suivant. L'algorithme fut introduit vers 532, par Dionysius Exiguus. Avec le calendrier Julien, (pour les années avant 1753), un cycle de 19 ans suffit pour connaître les dates des phases de la lune. Avec le calendrier grégorien, (à partir des années 1753, conçu par Clavius et Lilius, puis introduit par le pape Grégoire XIII en octobre 1582, et en Grande Bretagne et ses colonies en septembre 1752), deux facteurs de corrections ont été ajoutés pour rendre le cycle plus précis.

(Ce code est basé sur le programme en C de Simon Kershaw, <webmaster@ely.anglican.org>)

Voir easter_days pour les calculs de date de Pâques avant 1970 et après 2037.

8.8.11 easter_days() : Retourne le nombre de jours entre le 21 Mars et Pâques, pour une année donnée

int **easter_days** (int year , int method)

easter_days retourne le nombre de jours entre le 21 Mars et Pâques, pour une année donnée. Si l'année n'est pas précisée, l'année en cours est utilisée par défaut.

Depuis PHP 4.3.0, le paramètre year est optionnel, et prend la valeur de l'année en cours par défaut.

Le paramètre method a été introduit en PHP 4.3.0 et permet de calculer les dates de Pâques dans le calendrier Grégorien, entre les années 1582 - 1752, lorsque ce paramètre prend la valeur de CAL_EASTER_ROMAN . Voyez les constantes calendaires pour plus de constantes valides.

easter_days peut être utilisée à la place de easter_date pour calculer la date de Pâques, pour les années qui tombent hors de l'intervalle de validité des timestamps UNIX (i.e. avant 1970 ou après 2037).

Exemple avec easter_days

```
<?php
echo easter_days(1999);      /* 14, i.e. 4 Avril */
echo easter_days(1492);     /* 32, i.e. 22 Avril */
echo easter_days(1913);     /* 2, i.e. 23 Mars */

?>
```

La date de Pâques a été fixée par le concile de Nicée, en 325 de notre ère, comme étant le dimanche après la première pleine lune qui suit l'équinoxe de printemps. L'équinoxe de printemps

est considéré comme étant toujours le 21 mars, ce qui réduit le problème au calcul de la date de la lune pleine qui suit, et le dimanche suivant. L'algorithme fut introduit vers 532, par Dionysius Exiguus. Avec le calendrier Julien, (pour les années avant 1753), un cycle de 19 ans suffit pour connaître les date des phases de la lune. Avec le calendrier grégorien, (à partir des années 1753, conçu par Clavius et Lilius, puis introduit par le pape Grégoire XIII en octobre 1582, et en Grande Bretagne et ses colonies en septembre 1752), deux facteurs de corrections ont été ajoutés pour rendre le cycle plus précis.

(Ce code est basé sur le programme en C de Simon Kershaw, <webmaster@ely.anglican.org>)

Voir aussi [easter_date](#) .

8.8.12 FrenchToJD() : Convertit une date du calendrier français républicain en nombre de jours du calendrier Julien

int **frenchtojd** (int month , int day , int year)

FrenchToJD convertit une date du calendrier français républicain en nombre de jours du calendrier Julien.

Ces fonctions convertissent les dates comprises entre l'an 1 et l'an 14 (22 Septembre 1792 à 22 Septembre 1806 en calendrier grégorien). Cela couvre plus que la durée d'existence de ce calendrier.

8.8.13 GregorianToJD() : Convertit une date grégorienne en nombre de jours du calendrier Julien

int **gregoriantojd** (int month , int day , int year)

Intervalle de validité pour le calendrier grégorien : 4714 avant JC à 9999 après JC.A.D.

Bien qu'il soit possible de manipuler des dates jusqu'en 4714 avant JC, une telle utilisation n'est pas significative. En effet, ce calendrier fut créé le 18 octobre 1582 après J.C. (ou 5 octobre 1582 en calendrier grec). Certains pays ne l'acceptèrent que bien plus tard. Par exemple, les britanniques n'y passèrent qu'en 1752, les Russes en 1918 et les Grecs en 1923. La plupart des pays européens utilisaient le calendrier Julien avant le Grégorien.

Fonctions calendrier
<pre><?php \$jd = gregoriantojd(10,11,1970); echo "\$jd\n"; \$gregorian = jdtogregorian(\$jd); echo "\$gregorian\n"; ?></pre>

8.8.14 JDDayOfWeek() : Retourne le numéro du jour de la semaine

mixed **jddayofweek** (int julianday , int mode)

JDDayOfWeek retourne le numéro du jour de la semaine. Peut retourner une chaîne ou un entier, en fonction du mode.

8.8.11 **easter_days()** : Retourne le nombre de jours entre le 21 Mars et Pâques, pour une année 354 année

Mode	Signification
0 (défaut)	Retourne le numéro du jour comme un entier (0=dimanche, 1=lundi, etc.)
1	Retourne une chaîne contenant le nom du jour (anglais grégorien)
2	Retourne une chaîne contenant le nom abrégé du jour de la semaine (anglais grégorien).

8.8.15 JDMonthName() : Retourne le nom du mois

string **jdmonthname** (int julianday , int mode)

JDMonthName retourne une chaîne contenant le nom du mois. mode indique de quel calendrier dépend ce mois, et quel type de nom doit être retourné.

Mode	Signification	Valeurs
0	Grégorien - abrégé	Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec
1	Grégorien	January, February, March, April, May, June, July, August, September, October, November, December
2	Julien - abrégé	Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec
3	Julien	January, February, March, April, May, June, July, August, September, October, November, December
4	Juif	Tishri, Heshvan, Kislev, Tevet, Shevat, AdarI, AdarII, Nisan, Iyyar, Sivan, Tammuz, Av, Elul
5	Républicain français	Vendémiaire, Brumaire, Frimaire, Nivose, Pluviose, Ventose, Germinal, Floreal, Prairial, Messidor, Thermidor, Fructidor, Extra

8.8.16 JDToFrench() : Convertit le nombre de jours du calendrier Julien en date du calendrier français républicain

string **jdtofrench** (int juliandaycount)

JDToFrench convertit le nombre de jours du calendrier julien en date du calendrier français républicain.

8.8.17 JDToGregorian() : Convertit le nombre de jours du calendrier Julien en date grégorienne

string **jdtogregorian** (int julianday)

JDToGregorian convertit le nombre de jours du calendrier Julien en une chaîne contenant une date du calendrier grégorien, au format "mois/jour/année".

8.8.18 jdtojewish() : Convertit le nombre de jours du calendrier Julien en date du calendrier juif

string **jdtojewish** (int juliandaycount , bool hebrew , int fl)

jdtojewish convertit le nombre de jours du calendrier Julien en date du calendrier juif.

Les paramètres optionnels hebrew et fl sont disponibles depuis PHP 5.0.0.

Si le paramètre hebrew vaut TRUE , le paramètre fl sera utilisé pour générer une chaîne au format hébreux. Les formats disponibles sont : CAL_JEWISH_ADD_ALAFIM_GERESH , CAL_JEWISH_ADD_ALAFIM et CAL_JEWISH_ADD_GERESHAYIM .

Exemple avec `jdtojewish`

```
<?php
echo jdtojewish(gregoriantojd(10, 8, 2002), true,
                CAL_JEWISH_ADD_GERESHAYIM + CAL_JEWISH_ADD_ALAFIM + CAL_JEWISH_ADD_ALAFIM_GERESH);
?>
```

8.8.19 JDTToJulian() : Convertit le nombre de jours du calendrier Julien en date du calendrier Julien

string `jdtojulian` (int julianday)

`JDTToJulian` convertit le nombre de jours du calendrier Julien en une chaîne contenant la date du calendrier Julien, au format "mois/jour/année".

8.8.20 jdtounix() : Convertit un jours Julien en timestamp UNIX

int `jdtounix` (int jday)

`jdtounix` retourne un timestamp UNIX correspondant au jours Julien jday ou FALSE si jday n'est pas dans l'intervalle de validité de l'époque UNIX. (années grégoriennes entre 1970 et 2037 ou 2440588 <= jday <= 2465342). Le temps retourné est un temps local (et non GMT).

Voir aussi `unixtojd` .

8.8.21 JewishToJD() : Convertit une date du calendrier Juif en nombre de jours du calendrier Julien

int `jewishtojd` (int month , int day , int year)

Bien qu'il soit possible de manipuler des dates à partir de l'an 1 (3761 avant J.C.), une telle utilisation a peu de sens. Le calendrier juif a été utilisé depuis plusieurs dizaines de siècles, mais dans les premiers temps, il n'y avait pas de formule pour déterminer le début du mois. Un nouveau mois commençait quand une nouvelle lune était observée.

8.8.22 JulianToJD() : Convertit un jours du calendrier Julien en un nombre de jours du calendrier Julien

int `juliantojd` (int month , int day , int year)

Intervalle de validité du calendrier Julien : 4713 avant JC à 9999 après J.C..

Bien qu'il soit possible de manipuler des dates jusqu'en 4713 avant JC, une telle utilisation n'est pas significative. En effet, ce calendrier fut créé en 46 avant J.C., et ses détails ne furent finalisés qu'au

plus tôt en 8 après JC, et probablement pas avant le 4^{ème} siècle après JC. De plus, le début de l'année variait suivant les peuples, certains n'acceptant pas janvier comme premier mois de l'année.

Attention

Souvenez-vous, le calendrier courant du système utilisé sur le Web est un calendrier Grégorien. [GregorianToJD](#) peut être utilisé pour convertir ce genre de dates en un nombre de jours du calendrier Julien.

8.8.23 `unixtojd()` : Convertit un timestamp UNIX en un jours Julien

int `unixtojd` (*int timestamp*)

`unixtojd` retourne le jours Julien du timestamp UNIX timestamp (nombre de secondes depuis le 1/1/1970), ou pour le jour courant si timestamp est omis.

Voir aussi [jdtounix](#) .

8.9 Paiement CCVS [déconseillé]

8.9.1 Introduction

Ces fonctions font l'interface avec les API CCVS et vous permettant de travailler directement avec CCVS depuis vos scripts PHP. CCVS est la solution apportée par RedHat au problème de l'intermédiaire, lors du traitement de transactions de cartes de crédit. Il vous permet travailler directement avec les maisons de crédits, via votre boîte *nix et un modem. En utilisant le module CCVS pour PHP, vous pouvez effectuer des transactions avec les cartes de crédits, directement depuis vos scripts PHP via CCVS. La suite vous montrera comment procéder.

Note

CCVS a été abandonné par Red Hat et il n'y a pas de plan pour fournir de nouvelles clés ou du support. Ceux qui recherchent un système de remplacement peuvent regarder du côté de MCVE par Main Street Softworks . C'est une extension similaire à celle-ci, et qui dispose du support PHP.

Cette extension a été supprimée de PHP depuis la version 4.3.0 et n'est plus disponible. Si vous voulez utiliser des fonctionnalités similaires, utilisez MCVE .

8.9.2 Installation

Pour activer le support CCVS de PHP, commencez par vérifier votre installation CCVS. Vous devez configurer PHP avec l'option `--with-ccvs` . Si vous utilisez cette option sans spécifier le chemin de votre installation, PHP essaiera de la trouver à sa position par défaut (`/usr/local/ccvs`). Si CCVS est installé dans un autre dossier, lancez la configuration avec : `--with-ccvs=$ccvs_path` , où `$ccvs_path` est le chemin de votre installation CCVS. Notez bien que CCVS requiert que `$ccvs_path/lib` et `$ccvs_path/include` existent, et qu'ils contiennent respectivement `cv_api.h` et `libccvs.a` .

De plus, un démon `ccvsd` doit être disponible sur votre configuration, et qu'il soit accessible à vos scripts PHP. Assurez-vous aussi que l'utilisateur qui exécute les scripts PHP est le même que celui qui a installé CCVS (i.e. si vous avez installé CCVS avec l'utilisateur 'ccvs', vos scripts PHP doivent tourner aussi en 'ccvs').

8.9.3 Voir aussi

RedHat a arrêté le support de CCVS. Cependant, un manuel légèrement en retard est disponible à <http://www.redhat.com/docs/manuals/ccvs/> .

- [Introduction](#)
- [Installation](#)
- [Voir aussi](#)
- [ccvs_add](#)
- [ccvs_auth](#)
- [ccvs_command](#)
- [ccvs_count](#)
- [ccvs_delete](#)
- [ccvs_done](#)
- [ccvs_init](#)
- [ccvs_lookup](#)
- [ccvs_new](#)

- [ccvs_report](#)
- [ccvs_return](#)
- [ccvs_reverse](#)
- [ccvs_sale](#)
- [ccvs_status](#)
- [ccvs_textvalue](#)
- [ccvs_void](#)

8.9.5 **ccvs_auth()** : Effectue un test d'autorisation de crédit sur une transaction

string **ccvs_auth** (string session , string invoice)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.9.6 **ccvs_command()** : Exécute une commande spécifique à un protocole et qui n'est donc pas disponible dans l'API générique

string **ccvs_command** (string session , string type , string argval)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.9.7 **ccvs_count()** : Trouve le nombre de transactions d'un certain type stockées dans le système

int **ccvs_count** (string session , string type)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.9.8 **ccvs_delete()** : Supprime une transaction

string **ccvs_delete** (string session , string invoice)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.9.9 **ccvs_done()** : Termine l'enfin CCVS et libère les ressources

string **ccvs_done** (string sess)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.9.10 ccvs_init() : Initialise C CVS pour utilisation

string **ccvs_init** (string name)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.9.11 ccvs_lookup() : Recherche un élément d'un type spécifique dans la base de données

string **ccvs_lookup** (string session , string invoice , int inum)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.9.12 ccvs_new() : Crée une nouvelle transaction à vide

string **ccvs_new** (string session , string invoice)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.9.13 ccvs_report() : Retourne le status du processus de communication

string **ccvs_report** (string session , string type)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.9.14 ccvs_return() : Transfère des fonds de chez le marchand vers le possesseur de la carte de crédit

string **ccvs_return** (string session , string invoice)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.9.15 ccvs_reverse() : Annule entièrement une autorisation déjà donnée

string **ccvs_reverse** (string session , string invoice)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.9.16 **ccvs_sale()** : Transfère des fonds de chez le possesseur de la carte de crédit vers le marchand

string **ccvs_sale** (string session , string invoice)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.9.17 **ccvs_status()** : Vérifie le status d'une facture

string **ccvs_status** (string session , string invoice)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.9.18 **ccvs_textvalue()** : Retourne la valeur textuelle du dernier appel à une fonction

string **ccvs_textvalue** (string session)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.9.19 **ccvs_void()** : Annule complètement une transaction complétée

string **ccvs_void** (string session , string invoice)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.10 Fonctions Classkit

8.10.1 Introduction

Ces fonctions vous permettent de manipuler dynamiquement les classes PHP en cours de fonctionnement.

Note

Cette extension a été remplacée par [runkit](#), qui n'est pas limitée aux manipulations de classes mais peut également manipuler les fonctions.

8.10.2 Installation

Cette extension [PECL](#) n'est pas intégrée à PHP.

Des informations comme les notes sur les nouvelles versions, les téléchargements, les sources des fichiers, les informations concernant les mainteneurs ainsi qu'un CHANGELOG, peuvent être trouvées ici : <http://pecl.php.net/package/classkit>.

Vous pouvez télécharger ces bibliothèques DLL pour ces extensions PECL depuis la page [de téléchargement de PHP](#) sous sur la page <http://snaps.php.net/>.

8.10.3 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.10.4 Types de ressources

Cette extension ne définit aucune ressource.

8.10.5 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

`CLASSKIT_ACC_PRIVATE` ([entier](#))
Rend la méthode " private "
`CLASSKIT_ACC_PROTECTED` ([entier](#))
Rend la méthode " protected "
`CLASSKIT_ACC_PUBLIC` ([entier](#))
Rend la méthode " public "

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [classkit_import](#)

- [classkit_method_add](#)
- [classkit_method_copy](#)
- [classkit_method_redefine](#)
- [classkit_method_remove](#)
- [classkit_method_rename](#)

8.10.7 classkit_method_add() : Ajoute dynamiquement une nouvelle méthode à une classe donnée

bool **classkit_method_add** (string classname , string methodname , string args , string code , int flags)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

8.10.8 classkit_method_copy() : Copie une méthode d'une classe vers une autre classe

bool **classkit_method_copy** (string dClass , string dMethod , string sClass , string sMethod)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

8.10.9 classkit_method_redefine() : Change dynamiquement le code de la méthode donnée

bool **classkit_method_redefine** (string classname , string methodname , string args , string code , int flags)

Note

Cette fonction ne peut pas être utilisée pour manipuler la méthode en cours (ou liée) d'exécution.

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

8.10.10 classkit_method_remove() : Efface dynamiquement une méthode donnée

bool **classkit_method_remove** (string classname , string methodname)

Note

Cette fonction ne peut pas être utilisée pour manipuler la méthode en cours (ou liée) d'exécution.

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

8.10.11 `classkit_method_rename()` : Change dynamiquement le nom d'une méthode donnée

`bool classkit_method_rename (string classname , string methodname , string newname)`

Note

Cette fonction ne peut pas être utilisée pour manipuler la méthode en cours (ou liée) d'exécution.

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

8.11 Classes et Objets

8.11.1 Introduction

Ces fonctions vous donnent des informations sur les objets et les classes. Vous pouvez lire le nom de la classe qui est instanciée par un objet, ainsi que la liste des membres et des méthodes. En utilisant ces fonctions, vous pouvez non seulement connaître la classe d'un objet, mais aussi tous ses ascendants (i.e. la classe qui est étendue par une autre classe).

8.11.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.11.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.11.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.11.5 Types de ressources

Cette extension ne définit aucune ressource.

8.11.6 Constantes pré-définies

Cette extension ne définit aucune constante.

8.11.7 Exemples

Dans cet exemple, on définit une classe de base, et une extension. La classe de base définit un légume, s'il est mangeable ou pas et sa couleur. La sous-classe epinard ajoute une méthode pour le cuisiner, et une autre pour savoir s'il est cuisiné.

Fonctions d'objets : classes.inc

```
<?php
// classe de base, avec ses membres et ses méthodes
class Legume {
    var $edible;
    var $color;

    function Legume( $edible, $color="green" ) {
        $this->edible = $edible;
        $this->color = $color;
    }
}
```

```

    }

    function is_edible() {
        return $this->edible;
    }

    function what_color() {
        return $this->color;
    }
} // fin de la classe Legume

// Extension la classe de base
class Epinard extends Legume {
    var $cuit = FALSE;
    function Epinard() {
        $this->Legume( TRUE, "green" );
    }
    function cuisine() {
        $this->cuit = TRUE;
    }

    function is_cooked() {
        return $this->cooked;
    }
} // fin de la classe Epinard

?>

```

Lorsqu'on instancie deux objets de ces classes et que l'on affiche leurs informations, on affiche aussi leur héritage. On définit ici des utilitaires qui servent essentiellement à afficher ces informations proprement.

Fonctions d'objets : test_script.php

```

<pre>
<?php

include "classes.inc";

// Fonctions utilitaires
function print_vars($obj)
{
    foreach (get_object_vars($obj) as $prop => $val) {
        echo "\t$prop = $val\n";
    }
}

function print_methods($obj)
{
    $arr = get_class_methods(get_class($obj));
    foreach ($arr as $method) {
        echo "\tfunction $method()\n";
    }
}

function class_parentage($obj, $class) {
    if (is_subclass_of($GLOBALS[$obj], $class)) {
        echo "L'objet $obj appartient à la classe ".get_class($obj);
        echo " est une sous-classe de $class\n";
    } else {
        echo "L'objet $obj n'est pas une sous-classe de $class\n";
    }
}
}

```

```
// instantie 2 objets
$veggie = new Legume(true,"blue");
$leafy = new Epinard();

// affiche les informations sur ces objets
echo "légume : CLASS ".get_class($veggie)."\n";
echo "feuillues : CLASS ".get_class($leafy);
echo ", PARENT ".get_parent_class($leafy)."\n";

// affiche les propriétés du légume
echo "\nlégumes : Propriétés\n";
print_vars($veggie);

// et les méthodes des feuillus
echo "\nfeuillus : Méthodes\n";
print_methods($leafy);

    echo "\nParentée:\n";
class_parentage("leafy", "Epinard");
class_parentage("leafy", "Legume");
?>
</pre>
```

Il est important de noter que dans les exemples ci-dessus, les objets \$feuilles sont une instance de Epinard qui est une sous-classe de Legume , donc la dernière partie du script va afficher :

[...]

Parentée:

L'objet feuilles n'est pas une sous classe epinard

L'objet feuilles est une sous-classe de Legume

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [call_user_method_array](#)
- [call_user_method](#)
- [class_exists](#)
- [get_class_methods](#)
- [get_class_vars](#)
- [get_class](#)
- [get_declared_classes](#)
- [get_declared_interfaces](#)
- [get_object_vars](#)
- [get_parent_class](#)
- [interface_exists](#)
- [is_a](#)
- [is_subclass_of](#)
- [method_exists](#)
- [property_exists](#)

8.11.9 `call_user_method()` : Appelle une méthode utilisateur d'un objet

mixed **call_user_method** (string `method_name` , object `obj` , *mixed parameter* , *mixed ...*)

Attention

La fonction `call_user_method` est abandonnée depuis PHP 4.1.0, utilisez plutôt la fonction `call_user_func` avec la syntaxe `array(&$obj, "method_name")` .

`call_user_method` appelle la méthode `method_name` de l'objet `obj` , avec les paramètres `parameter` , Un exemple d'utilisation de cet objet est présenté ci-dessous, où une classe est définie, puis instantiée. On utilise alors `call_user_method` pour appeler indirectement les méthodes `print_info` .

Exemple avec `call_user_method`

```

<?php
class Country {
    var $NAME;
    var $TLD;

    function Country($name, $tld) {
        $this->NAME = $name;
        $this->TLD = $tld;
    }

    function print_info($prestr="") {
        echo $prestr."Pays : ".$this->NAME."\n";
        echo $prestr."Top Level Domain : ".$this->TLD."\n";
    }
}

$cntry = new Country("Perou", "pe");

echo "* Appel de la méthode de l'objet directement\n";
$cntry->print_info();

echo "\n* Appel de la méthode de l'objet indirectement\n";
call_user_method ("print_info", $cntry, "\t");
?>

```

Voir aussi `call_user_func_array` et `call_user_func` .

8.11.10 `class_exists()` : Vérifie qu'une classe a été définie

bool **class_exists** (string `class_name` , *bool autoload*)

`class_exists` retourne TRUE si la classe `class_name` a été définie, et FALSE sinon.

Exemple avec `class_exists`

```

<?php
// Vérifiez que la classe existe avant de l'utiliser
if (class_exists('MaClasse')) {
    $maclasse = new MaClasse();
}
?>

```

`class_exists` essaiera d'appeler `__autoload` par défaut. Si vous ne le voulez pas, vous pouvez définir le paramètre `autoload` à FALSE .

Exemple avec le paramètre autoload

```

<?php
function __autoload($class) {
    include($class . '.php');

    /* Vérifie si l'include définit la classe */
    if ( !class_exists($class, false) ) {
        trigger_error("Impossible de charger la classe $class", E_USER_ERROR);
    }
}

if (class_exists('MaClasse')) {
    $maclasse = new MaClasse();
}
?>

```

Note

Le paramètre autoload a été ajouté en PHP 5.

Voir aussi [get_declared_classes](#) et [interface_exists](#) .

8.11.11 get_class_methods() : Retourne les noms des méthodes d'une classe

array **get_class_methods** (mixed class_name)

get_class_methods retourne un tableau contenant les noms des méthodes de la classe class_name . En cas d'erreur, NULL est retourné.

Note

A partir de PHP 4.0.6, vous pouvez spécifier l'objet lui-même, au lieu de sa classe class_name . Par exemple :

Passer l'objet ou la classe à get_class_methods

```

<?php
$class_methods = get_class_methods($my_class);
?>

```

Exemple avec get_class_methods

```

<?php
class maclasse {
    // constructeur
    function maclasse() {
        return(TRUE);
    }
    // méthode 1
    function myfunc1() {
        return(TRUE);
    }
    // méthode 2
    function mafunc2() {
        return(TRUE);
    }
}

$class_methods = get_class_methods('my_object');

```

```
foreach ($class_methods as $method_name) {
    echo "$method_name\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
maclasse
myfunc1
mafunc2
```

Attention

Depuis PHP 5, cette fonction retourne le nom des méthodes comme elles ont été déclarées (sensible à la casse). En PHP 4, elles étaient converties en minuscule.

Voir aussi [get_class](#) , [get_class_vars](#) et [get_object_vars](#) .

8.11.12 `get_class_vars()` : Retourne les valeurs par défaut des propriétés d'une classe

array `get_class_vars` (string class_name)

`get_class_vars` retourne un tableau contenant les valeurs par défaut des propriétés publiques de la classe `class_name` .

Note

Avant PHP 4.2.0, les variables de classe non initialisées n'étaient pas rapportées par `get_class_vars` .

Exemple avec `get_class_vars`

```
<?php

class myclass {

    var $var1; // this has no default value...
    var $var2 = "xyz";
    var $var3 = 100;
    private $var4; // PHP 5

    // constructeur
    function myclass() {
        return(TRUE);
    }
}

$my_class = new myclass();

$class_vars = get_class_vars(get_class($my_class));

foreach ($class_vars as $name => $value) {
    echo "$name : $value\n";
}

?>
```

L'exemple ci-dessus va afficher :

```
// Avant PHP 4.2.0
var2 : xyz
var3 : 100

// Depuis PHP 4.2.0
var1 :
var2 : xyz
var3 : 100
```

Voir aussi [get_class_methods](#) et [get_object_vars](#) .

8.11.13 `get_class()` : Retourne la classe d'un objet

string `get_class` (*object* *obj*)

`get_class` retourne la classe de l'objet *obj* . `get_class` retourne FALSE si *obj* n'est pas un objet.

Note

`get_class` retourne un nom de classe utilisateur, en minuscules. Une classe définie par une extension PHP est retournée dans sa notation originale.

Note

Depuis PHP 5, *obj* est optionnel s'il est appelé depuis la méthode de l'objet.

Exemple avec `get_class`

```
<?php
class foo {
    function foo() {
        // un peu de code
    }

    function name() {
        echo "Mon nom est " , get_class($this) , "\n";
    }
}

// create an object
$bar = new foo();

// external call
echo "Son nom est " , get_class($bar) , "\n";

// internal call
$bar->name();

?>
```

L'exemple ci-dessus va afficher :

```
Son nom est foo
Mon nom est foo
```

Voir aussi [get_parent_class](#) , [gettype](#) et [is_subclass_of](#) .

8.11.14 `get_declared_classes()` : Liste toutes les classes définies

array `get_declared_classes` (void)

`get_declared_classes` retourne un tableau contenant la liste des classes déclarées dans le script courant.

Note

En PHP 4.0.1pl2, trois classes supplémentaires sont retournées, au début de ce tableau : `stdClass` (définie dans `Zend/zend.c`), `OverloadedTestClass` (définie dans `ext/standard/basic_functions.c`) et `Directory` (définie dans `ext/standard/dir.c`).

De plus, notez que suivant les bibliothèques qui sont compilées avec PHP, d'autres classes peuvent être présentes. Cela signifie que vous ne pourrez pas utiliser ces noms de classe pour définir vos propres classes. Voici une liste des classes pré-définies .

Exemple avec `get_declared_classes`

```
<?php
print_r(get_declared_classes());
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[0] => stdClass
[1] => __PHP_Incomplete_Class
[2] => Directory
)
```

Voir aussi [class_exists](#) et [get_declared_interfaces](#) .

8.11.15 `get_declared_interfaces()` : Retourne un tableau avec toutes les interfaces déclarées

array `get_declared_interfaces` (void)

`get_declared_interfaces` retourne un tableau avec la liste de toutes les interfaces déclarées dans le script courant.

Exemple avec `get_declared_interfaces`

```
<?php
print_r(get_declared_interfaces());
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[0] => Traversable
)
```

```
[1] => IteratorAggregate
[2] => Iterator
[3] => ArrayAccess
[4] => reflector
[5] => RecursivIterator
[6] => SeekableIterator
)
```

Voir aussi [get_declared_classes](#) .

8.11.16 `get_object_vars()` : Retourne un tableau associatif des propriétés d'un objet

array `get_object_vars` (object obj)

`get_object_vars` retourne un tableau associatif contenant les propriétés de l'objet obj . Les clés du tableau sont les noms des propriétés de l'objet.

Note

Dans les versions antérieures à la PHP 4.2.0, si les variables déclarées dans la classe de l'objet obj , n'avaient pas été assignées, elles n'apparaissent pas dans le tableau retourné. Depuis PHP 4.2.0, elles apparaissent, avec la valeur NULL .

Exemple avec `get_object_vars`

```
<?php
class Point2D {
    var $x, $y;
    var $label;

    function Point2D($x, $y) {
        $this->x = $x;
        $this->y = $y;
    }

    function setLabel($label) {
        $this->label = $label;
    }

    function getPoint() {
        return array("x" => $this->x,
                    "y" => $this->y,
                    "label" => $this->label);
    }
}

// "$label" est déclaré mais non défini
$p1 = new Point2D(1.233, 3.445);
print_r(get_object_vars($p1));

$p1->setLabel("point #1");
print_r(get_object_vars($p1));

?>
```

L'exemple ci-dessus va afficher :

```

Array
(
[x] => 1.233
[y] => 3.445
[label] =>
)

Array
(
[x] => 1.233
[y] => 3.445
[label] => point #1
)

```

Voir aussi [get_class_methods](#) et [get_class_vars](#) .

8.11.17 [get_parent_class\(\)](#) : Retourne le nom de la classe d'un objet

string [get_parent_class](#) (*mixed* *obj*)

[get_parent_class](#) retourne le nom de la classe de l'objet *obj* .

Si le paramètre *obj* est une chaîne, [get_parent_class](#) retourne le nom de la classe parente. Cette fonctionnalité a été ajoutée en PHP 4.0.5.

Note

Depuis PHP 5, *obj* est optionnel s'il est appelé depuis la méthode de l'objet.

Exemple avec [get_parent_class](#)

```

<?php

class papa {
    function papa() {
        // un peu de code
    }
}

class child extends papa {
    function child() {
        echo "je suis le fils de " , get_parent_class($this) , "\n";
    }
}

class child2 extends papa {
    function child2() {
        echo "Je suis aussi le papa de " , get_parent_class('child2') , "\n";
    }
}

$foo = new child();
$bar = new child2();

?>

```

L'exemple ci-dessus va afficher :

```
je suis le fils de papa
Je suis aussi le papa de papa
Voir aussi get\_class et is\_subclass\_of .
```

8.11.18 `interface_exists()` : Vérifie si une interface a été définie

`bool interface_exists (string interface_name , bool autoload)`

`interface_exists` retourne TRUE si l'interface `interface_name` a été définie, et FALSE sinon.

Exemple avec `interface_exists`

```
<?php
// Vérifie si l'interface existe avant de l'utiliser
if (interface_exists('MonInterface')) {
    class MaClasse implements MonInterface
    {
        // Méthodes
    }
}
?>
```

`interface_exists` essaiera d'appeler `__autoload` par défaut. Si vous ne voulez pas que `interface_exists` appelle `__autoload`, vous pouvez donner au paramètre `autoload` la valeur de FALSE .

Voir aussi [class_exists](#) .

8.11.19 `is_a()` : Retourne TRUE si un objet a pour parent une classe donnée

`bool is_a (object object , string class_name)`

`is_a` retourne TRUE si l'objet `object` a pour classe `class_name`, ou bien si `class_name` est un de ses parents. `is_a` retourne FALSE sinon.

Exemple avec `is_a`

```
<?php
// Définit une classe
class WidgetFactory
{
    var $oink = 'moo';
}

// Crée un nouvel objet
$WF = new WidgetFactory();

if (is_a($WF, 'WidgetFactory')) {
    echo "oui, \$WF est toujours un objet WidgetFactory\n";
}
?>
```

La fonction `is_a` est déconseillé en PHP 5 en faveur de l'opérateur de type `instanceof` . Voir un exemple de cette utilisation avec PHP 5 :

Utilisation de l'opérateur **instanceof** avec PHP 5

```
<?php
if ($WF instanceof WidgetFactory) {
    echo 'Oui, $WF est un WidgetFactory';
}
?>
```

Voir aussi [get_class](#) , [get_parent_class](#) et [is_subclass_of](#) .

8.11.20 is_subclass_of() : Détermine si un objet est une sous-classe

bool **is_subclass_of** (mixed object , string class_name)

[is_subclass_of](#) retourne TRUE si l'objet object est une sous-classe de class_name , FALSE sinon.

Note

Depuis PHP 5.0.3, vous devez également spécifier le paramètre object en tant que chaîne de caractères (le nom de la classe).

Exemple avec [is_subclass_of](#)

```
<?php
// définition d'une classe
class WidgetFactory
{
    var $oink = 'moo';
}

// définition d'une sous-classe
class WidgetFactory_Child extends WidgetFactory
{
    var $oink = 'oink';
}

// création d'un nouvel objet
$WF = new WidgetFactory();
$WFC = new WidgetFactory_Child();

if (is_subclass_of($WFC, 'WidgetFactory')) {
    echo "oui, \$WFC est une sous-classe de la classe WidgetFactory\n";
} else {
    echo "non, \$WFC n'est pas une sous-classe de la classe WidgetFactory\n";
}

if (is_subclass_of($WF, 'WidgetFactory')) {
    echo "oui, \$WF est une sous-classe de la classe WidgetFactory\n";
} else {
    echo "non, \$WF n'est pas une sous-classe de la classe WidgetFactory\n";
}

// utilisable uniquement depuis PHP 5.0.3
if (is_subclass_of('WidgetFactory_Child', 'WidgetFactory')) {
    echo "oui, WidgetFactory_Child est une sous-classe de la classe WidgetFactory\n";
} else {
    echo "non, WidgetFactory_Child n'est pas une sous-classe de la classe WidgetFactory\n";
}
?>
```

L'exemple ci-dessus va afficher :

oui, \$WFC est une sous-classe de la classe WidgetFactory
 non, \$WF n'est pas une sous-classe de la classe WidgetFactory
 oui, WidgetFactory_Child est une sous-classe de la classe WidgetFactory

Voir aussi [get_class](#) et [get_parent_class](#) .

8.11.21 `method_exists()` : Vérifie que la méthode existe pour une classe

bool **method_exists** (object object , string method_name)

`method_exists` retourne TRUE si la méthode `method_name` a été définie pour la classe `object` , et sinon, retourne FALSE .

Exemple avec `method_exists`

```
<?php
$directory = new Directory('.');
var_dump(method_exists($directory, 'read'));
?>
```

L'exemple ci-dessus va afficher :

```
bool(true)
```

Voir aussi [function_exists](#) et [is_callable](#) .

8.11.22 `property_exists()` : Vérifie si un objet ou une classe a une propriété

bool **property_exists** (mixed class , string property)

Cette fonction vérifie si la propriété `property` existe dans la classe spécifiée (et si elle est déclarée comme publique).

Note

À l'opposé de la fonction [isset](#) , `property_exists` retourne TRUE même si la propriété a une valeur NULL .

8.12 COM et .Net (Windows)

8.12.1 Introduction

COM est un acronyme pour Component Object Model; c'est une couche orientée objet (et services associés) au-dessus de DCE RPC (un standard libre) qui définit une convention de nommage commune qui permet à un code écrit en n'importe quel langage de communiquer avec du code écrit dans un autre langage, sous peu que ces deux langages soient compatibles avec COM. Les codes n'ont pas besoin de faire partie du même exécutable. Le code peut être chargé à partir d'une DLL, trouvé dans un autre processus fonctionnant sur le même serveur, ou, avec DCOM (Distributed COM), trouvé sur une machine distante, sans même que votre code ait à savoir où il se trouve.

Il y'a une partie de COM connue sous le nom de OLE Automation qui comprend un jeu d'interfaces COM qui permettent de perdre l'attache aux objets COM, pour qu'ils puissent être introspectés et appelés en cours d'exécution sans savoir au moment de la compilation comment l'objet fonctionne. L'extension COM de PHP utilise les interfaces OLE Automation pour vous permettre de créer et appeler des objets compatibles depuis vos scripts. Techniquement parlant, cela devrait vraiment s'appeler " the OLE Automation Extension for PHP ", puisque tous les objets COM ne sont pas forcément compatibles OLE.

Maintenant, pourquoi voudriez-vous ou devriez-vous utiliser COM ? COM est l'une des méthodes les plus utilisées pour faire communiquer des applications et des composants sur les plates-formes Windows. En utilisant COM, vous pouvez ouvrir un document Microsoft Word, remplir un fichier de template et sauvegarder pour l'envoyer par email à votre visiteur. Vous pouvez aussi utiliser COM pour effectuer des tâches administratives sur votre réseau et configurer IIS; ce ne sont que les utilisations les plus habituelles, vous pouvez faire beaucoup plus avec COM.

Avec PHP 5, cette extension (et cette documentation) ont été réécrites de A à Z et la plupart des vieux problèmes et concepts portant à confusion ont été supprimés. De plus, nous supportons l'instanciation et la création d'assemblées .Net utilisant une couche d'interopérabilité COM fournie par Microsoft.

Lisez bien [cet article](#) pour un aperçu des modifications de cette extension dans PHP 5.

8.12.2 Pré-requis

Les fonctions COM sont uniquement disponibles pour les versions Windows de PHP.

Le support .NET requiert PHP 5 et l'exécutable .NET.

8.12.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de bibliothèque supplémentaire pour disposer de ces fonctions.

Vous êtes responsables de l'installation du support des divers objets COM que vous voulez utiliser (comme MS Word); nous ne pouvons et ne voulons pas les inclure tous avec PHP.

8.12.4 For Each

Avec PHP 5, vous pouvez utiliser la structure de contrôle [foreach](#) de PHP pour itérer à travers le contenu d'un IEnumVariant COM/OLE standard. Cela signifie que vous pouvez utiliser foreach aux endroits où vous auriez pu utiliser For Each dans du code VB/ASP.

For Each en ASP

```
<%
Set domainObject = GetObject("WinNT://Domain")
For Each obj in domainObject
    Response.Write obj.Name & "<br />"
Next
%>
```

while() ... Next() en PHP 4

```
<?php
$domainObject = new COM("WinNT://Domain");
while ($obj = $domainObject->Next()) {
    echo $obj->Name . "<br />";
}
?>
```

foreach en PHP 5

```
<?php
$domainObject = new COM("WinNT://Domain");
foreach ($domainObject as $obj) {
    echo $obj->Name . "<br />";
}
?>
```

8.12.5 Tableaux et propriétés à la façon des tableaux de COM

Plusieurs objets COM exposent leurs propriétés comme tableaux, ou en utilisant un chemin d'accès à la façon des tableaux. En PHP 4, vous pouvez utiliser la syntaxe de tableaux de PHP pour lire/écrire une propriété de ce genre, mais une seule dimension est permise. Si vous voulez lire une propriété multi-dimensionnelle, vous pouvez accéder à la propriété à travers une fonction ou chaque paramètre représente une dimension de ce tableau, mais il n'y a aucun moyen d'écrire une telle propriété.

PHP 5 introduit différentes nouvelles fonctionnalités pour vous faciliter la vie:

- Accédez aux tableaux multi-dimensionnels ou aux propriétés COM qui requièrent plusieurs paramètres comme si vous accédez à un tableau. Vous pouvez aussi écrire ces propriétés en utilisant cette technique.
- Bouclez sur les SafeArrays ("vrais" tableaux) en utilisant la structure de contrôle [foreach](#). Cela fonctionne car un SafeArrays comportent des informations à propos de sa taille. Si une propriété à la façon des tableaux implémente IEnumVariant alors vous pouvez aussi utiliser foreach pour cette propriété; lisez [For Each](#) pour plus d'informations à ce sujet.

8.12.6 Exceptions (PHP 5)

Cette extension lancera des instances de la classe `com_exception` pour toute erreur fatale reportée par COM. Toutes les exceptions COM ont une propriété `code` qui correspond à la valeur de retour

HRESULT des diverses opérations COM. Vous pouvez utiliser ce code pour choisir de façon automatique comment gérer cette exception.

8.12.7 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable	Historique
<code>com.allow_dcom</code>	"0"	PHP_INI_SYSTEM	Disponible depuis PHP 4.0.5.
<code>com.autoregister_typelib</code>	"0"	PHP_INI_ALL	PHP_INI_SYSTEM en PHP 4. Disponible depuis PHP 4.1.0.
<code>com.autoregister_verbose</code>	"0"	PHP_INI_ALL	PHP_INI_SYSTEM en PHP 4. Disponible depuis PHP 4.1.0.
<code>com.autoregister_casesensitive</code>	"1"	PHP_INI_ALL	PHP_INI_SYSTEM en PHP 4. Disponible depuis PHP 4.1.0.
<code>com.code_page</code>	""	PHP_INI_ALL	Disponible depuis PHP 5.0.0.
<code>com.typelib_file</code>	""	PHP_INI_SYSTEM	Disponible depuis PHP 4.0.5.

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du `php.ini`](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

`com.allow_dcom`

Si cette directive est activée, PHP aura la permission d'opérer comme un client D-COM (Distributed COM) et permettra à PHP d'instancier des objets COM sur un serveur distant.

`com.autoregister_typelib`

Si cette directive est activée, PHP essaiera de déclarer des constantes provenant de la bibliothèque `typelibrary` des objets qu'il instancie, si ces objets implémentent l'interface demandée pour obtenir les données demandées. La sensibilité des constantes à la casse est contrôlée par la directive de configuration [com.autoregister_casesensitive](#).

`com.autoregister_verbose`

Quand cette directive est activée, tout problème rencontré lors du chargement d'une `typelibrary` durant l'instanciation de l'objet sera reportée en utilisant le mécanisme d'erreurs de PHP. Par défaut, elle est désactivée, ce qui ne donne aucune indication sur l'échec de l'opération lors d'une recherche ou d'un chargement de la bibliothèque de type.

`com.autoregister_casesensitive`

Cette directive est activée par défaut et fait que les constantes trouvées dans les bibliothèques de types seront enregistrées en mode sensible à la casse. Voyez [com_load_typelib](#) pour plus de détails.

`com.code_page`

Cette directive permet de spécifier le code-page des jeux de caractères à utiliser lors de l'envoi et la réception de chaînes vers des objets COM. Si elle est vide, PHP pensera que vous voulez `CP_ACP`, qui est le code page système ANSI par défaut.

Si le texte dans vos scripts est encodé avec un différent encodage ou jeu de caractères par défaut, configurer cette directive vous évitera d'avoir à passer tout votre code comme paramètre du constructeur de la classe `COM`. Notez qu'en utilisant cette directive (comme toute configuration de PHP), votre code PHP perd en portabilité. Vous devez utiliser le paramètre du constructeur à chaque fois que

possible.

Note

Cette directive a été introduite en PHP 5.
--

`com.typelib_file`

Lorsqu'elle est configurée, cette directive doit être le chemin vers un fichier qui contient une liste de bibliothèques à charger au démarrage. Chaque ligne sera interprétée comme le nom de la bibliothèque de types et chargée comme si vous aviez utilisé `com_load_typelib`. Les constantes seront enregistrées de façon persistante, pour que la bibliothèque ne soit chargée qu'une seule fois. Si le nom d'une bibliothèque de types se termine par `#cis` ou `#case_insensitive`, alors les constantes de cette bibliothèque de types sera enregistrée en mode insensible à la casse.

8.12.8 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

`CLSCTX_INPROC_SERVER` ([entier](#))
`CLSCTX_INPROC_HANDLER` ([entier](#))
`CLSCTX_LOCAL_SERVER` ([entier](#))
`CLSCTX_REMOTE_SERVER` ([entier](#))
`CLSCTX_SERVER` ([entier](#))
`CLSCTX_ALL` ([entier](#))
`VT_NULL` ([entier](#))
`VT_EMPTY` ([entier](#))
`VT_UI1` ([entier](#))
`VT_I2` ([entier](#))
`VT_I4` ([entier](#))
`VT_R4` ([entier](#))
`VT_R8` ([entier](#))
`VT_BOOL` ([entier](#))
`VT_ERROR` ([entier](#))
`VT_CY` ([entier](#))
`VT_DATE` ([entier](#))
`VT_BSTR` ([entier](#))
`VT_DECIMAL` ([entier](#))
`VT_UNKNOWN` ([entier](#))
`VT_DISPATCH` ([entier](#))
`VT_VARIANT` ([entier](#))
`VT_I1` ([entier](#))
`VT_UI2` ([entier](#))
`VT_UI4` ([entier](#))
`VT_INT` ([entier](#))
`VT_UINT` ([entier](#))
`VT_ARRAY` ([entier](#))
`VT_BYREF` ([entier](#))
`CP_ACP` ([entier](#))
`CP_MACCP` ([entier](#))
`CP_OEMCP` ([entier](#))
`CP_UTF7` ([entier](#))
`CP_UTF8` ([entier](#))

CP_SYMBOL ([entier](#))
CP_THREAD_ACP ([entier](#))
VARCMP_LT ([entier](#))
VARCMP_EQ ([entier](#))
VARCMP_GT ([entier](#))
VARCMP_NULL ([entier](#))
NORM_IGNORECASE ([entier](#))
NORM_IGNORENONSPACE ([entier](#))
NORM_IGNORESYMBOLS ([entier](#))
NORM_IGNOREWIDTH ([entier](#))
NORM_IGNOREKANATYPE ([entier](#))
NORM_IGNOREKASHIDA ([entier](#))
DISP_E_DIVBYZERO ([entier](#))
DISP_E_OVERFLOW ([entier](#))
MK_E_UNAVAILABLE ([entier](#))

8.12.9 Voir aussi

Pour plus d'informations sur les objets COM, lisez les [spécifications COM](#) ou bien jetez un oeil au livre de Don Box (en anglais) : [Yet Another COM Library \(YACL\)](#) . Vous pouvez trouver des informations utiles dans notre FAQ pour [PHP et COM](#) . Si vous voulez utiliser des applications MS Office sur le serveur, vous devez lire les informations ici : [Considerations for Server-Side Automation of Office](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [For Each](#)
- [Tableaux et propriétés à la façon des tableaux de COM](#)
- [Exceptions \(PHP 5\)](#)
- [Configuration à l'exécution](#)
- [Constantes pré-définies](#)
- [Voir aussi](#)
- [COM](#)
- [DOTNET](#)
- [VARIANT](#)
- [com_addrf](#)
- [com_create_guid](#)
- [com_event_sink](#)
- [com_get_active_object](#)
- [com_get](#)
- [com_invoke](#)
- [com_isenum](#)
- [com_load_typelib](#)
- [com_load](#)
- [com_message_pump](#)
- [com_print_typeinfo](#)
- [com_propget](#)
- [com_propput](#)
- [com_propset](#)
- [com_release](#)
- [com_set](#)
- [variant_abs](#)

- [variant_add](#)
- [variant_and](#)
- [variant_cast](#)
- [variant_cat](#)
- [variant_cmp](#)
- [variant_date_from_timestamp](#)
- [variant_date_to_timestamp](#)
- [variant_div](#)
- [variant_eqv](#)
- [variant_fix](#)
- [variant_get_type](#)
- [variant_idiv](#)
- [variant_imp](#)
- [variant_int](#)
- [variant_mod](#)
- [variant_mul](#)
- [variant_neg](#)
- [variant_not](#)
- [variant_or](#)
- [variant_pow](#)
- [variant_round](#)
- [variant_set_type](#)
- [variant_set](#)
- [variant_sub](#)
- [variant_xor](#)

8.12.11 DOTNET() : La classe DOTNET

```
$obj = new DOTNET ("assembly", "classname")
```

8.12.12 VARIANT() : classe VARIANT

```
$vVar = new VARIANT ($var)
```

8.12.13 com_addref() : Incrmente le compteur de références (déconseillé)

```
void com_addref ( void )
```

[com_addref](#) incrémente le compteur de références.

Attention
Vous ne devriez jamais avoir besoin d'utiliser cette fonction.
Note
Cette fonction a été supprimé de PHP 5.

8.12.14 com_create_guid() : Génère un identifiant unique global

```
string com_create_guid ( void )
```

`com_create_guid` génère un identifiant unique global (GUID) et le retourne sous forme d'une chaîne. Un GUID est généré de la même façon que DCE UUID, excepté le fait que la convention Microsoft enclot le GUID dans une parenthèse.

Voir aussi `uuid_create` dans l'extension PECL uuid.

8.12.15 `com_event_sink()` : Connecte des événements d'un objet COM sur un objet PHP

`bool com_event_sink (variant comobject , object sinkobject , mixed sinkinterface)`

`com_event_sink` connecte des événements de l'objet COM `comobject` sur un l'objet PHP `sinkobject`. PHP devrait être capable d'utiliser le type par défaut de `dispinterface` spécifié par la Typelib associée avec l'objet `comobject`, mais vous pouvez changer cela en spécifiant dans le paramètre `sinkinterface` le `dispinterface` que vous souhaitez utiliser.

`sinkobject` devra être une instance de la classe avec des noms de méthodes suivant le `dispinterface` désiré ; vous devriez utilisé `com_print_typeinfo` pour l'aide à la génération d'un gabarit de classe pour cela.

Soyez prudent sur l'utilisation de cette fonctionnalité ; si vous faites quelque chose de similaire à l'exemple ci-dessous, cela n'a pas de sens de le lancer sur un serveur Web.

Exemple de connexions d'événements COM

```
<?php
class IEEventSinker {
    var $terminated = false;

    function ProgressChange($progress, $progressmax) {
        echo "Progression du téléchargement : $progress / $progressmax\n";
    }

    function DocumentComplete(&$dom, $url) {
        echo "Document $url terminé\n";
    }

    function OnQuit() {
        echo "Quitter!\n";
        $this->terminated = true;
    }
}

$ie = new COM("InternetExplorer.Application");
// notez que vous n'avez pas besoin du & en PHP 5!
$sink =& new IEEventSinker();
com_event_sink($ie, $sink, "DWebBrowserEvents2");
$ie->Visible = true;
$ie->Navigate("http://www.php.net");
while(!$sink->terminated) {
    com_message_pump(4000);
}
$ie = null;
?>
```

Voir aussi `com_print_typeinfo` et `com_message_pump`.

8.12.16 `com_get_active_object()` : Retourne un objet représentant l'instance actuelle d'un objet COM

variant `com_get_active_object` (string `progid` , int `code_page`)

`com_get_active_object` est similaire à la création d'une nouvelle instance COM d'un objet COM, excepté qu'elle ne retournera un objet à votre script uniquement si l'objet est actuellement instancié. Les applications OLE utilisent quelque chose connue en tant que " Running Object Table " qui permet aux applications connues d'être exécutées juste une fois ; cette fonction expose la fonction `GetActiveObject()` de la bibliothèque COM pour récupérer un objet d'une instance en cours d'utilisation.

Le paramètre `Progid` doit être soit le `ProgID`, soit le `CLSID` de l'objet dont vous voulez avoir l'accès (par exemple, `Word.Application`). Le paramètre `code_page` utilise les mêmes règles que dans la COM classe.

Si l'objet demandé est en cours d'exécution, la fonction `com_get_active_object` retournera à votre script ce que toute autre objet COM retournerait. Sinon, un `com_exception` sera augmenté. Il y a beaucoup de raisons pour lesquelles cette fonction peut échouer. Dans cette situation, le code erreur de l'exception devrait être `MK_E_UNAVAILABLE` ; vous pouvez utiliser la méthode `getCode` de l'objet exception pour vérifier le code de l'exception.

Attention

Utiliser la fonction `com_get_active_object` sur un serveur web n'est pas toujours la meilleure idée. La plus part des applications COM/OLE ne sont pas faites pour gérer plus d'un client en concurrence, comme (et surtout !!) Microsoft Office. Vous devriez lire les considérations pour les automatisms coté serveur pour Office pour plus d'informations sur les comportements généraux.

8.12.17 `com_get()` : Lit la valeur d'une propriété d'un composant COM

mixed `com_get` (resource `com_object` , string `property`)

`com_get` retourne la valeur de la propriété `property` du composant COM `com_object` . `com_get` retourne `FALSE` en cas d'erreur.

N'utiliser pas `com_get()`, utilisez la syntaxe OO à la place

```
<?php
// do this
$var = $obj->property;
// instead of this:
$var = com_get($obj, 'property');
?>
```

Note

Cette fonction n'existe pas en PHP 5 ; à la place, vous devriez utiliser la syntaxe régulière et plus naturelle OO pour accéder aux propriétés ou appeler les méthodes.

8.12.18 `com_invoke()` : Appelle une méthode d'un composant (déconseillé)

mixed `com_invoke` (resource `com_object` , string `function_name` , mixed `function_parameters`)

`com_invoke` appelle la méthode fonction_name du composant COM com_object . `com_invoke` retourne FALSE en cas d'erreur, sinon retourne le résultat de la fonction fonction_name en cas de succès. Tous les paramètres supplémentaires fonction_parameters sont passés à la méthode fonction_name .

N'utiliser pas `com_invoke()`, utilisez la syntaxe OO à la place

```
<?php
// do this
$val = $obj->method($one, $two);
// instead of this:
$val = com_invoke($obj, 'method', $one, $two);
?>
```

Note

Cette fonction n'existe pas en PHP 5 ; à la place, vous devriez utiliser la syntaxe régulière et plus naturelle OO pour accéder aux propriétés ou appeler les méthodes.

8.12.19 com_isenum() : Récupère un IEnumVariant (déconseillé)

bool **com_isenum** (variant com_module)

Regarde si l'objet COM peut être énuméré en utilisant la méthode (hack) Next() . Retourne TRUE s'il le peut, FALSE sinon. Voir [COM](#) la classe pour plus de détails sur ces méthodes.

Note

Cette fonction n'existe pas en PHP 5 qui utilise une façon plus naturelle pour instancier le contenu des objets COM. Voir [For Each](#) pour plus de détails.

8.12.20 com_load_typelib() : Charge un Typelib

bool **com_load_typelib** (string typelib_name , bool case_insensitive)

Charge un Typelib et enregistre son contenu dans le moteur, comme s'ils avaient été définis en utilisant [define](#) . Le paramètre case_insensitive se comporte comme le paramètre du même nom de la fonction [define](#) .

typelib_name peut être une des valeurs suivantes :

- Le nom d'un fichier .tlb ou le module exécutable contenant le Typelib.
- Le GUID du Typelib, suivi du numéro de version ; par exemple {00000200-0000-0010-8000-00AA006D2EA4},2,0 .
- Le nom du Typelib, e.g Microsoft OLE DB ActiveX Data Objects 1.0 Library .

PHP va essayer de résoudre le Typelib dans cet ordre, et le processus prendra de plus en plus de ressources pour arriver à la fin de la liste ; la recherche du Typelib par son nom est manipulé physiquement en énumérant le registre jusqu'à ce qu'un résultat y soit trouvé.

Notez qu'il est plus efficace d'utiliser [com.typelib_file](#) l'option de configuration pour précharger et enregistrer les constantes, même si cela est moins flexible.

Si vous devez activer [com.autoregister_typelib](#) , alors PHP essayera d'enregistrer automatiquement les constantes associées à l'objet COM lorsque vous l'instanciez. Cela dépendra de l'interface fourni par l'objet COM lui-même et ne sera pas toujours possible.

8.12.21 `com_load()` : Crée une référence sur un composant COM (déconseillé)

resource **com_load** (string *module_name* , string *server_name* , int *codepage*)

Produit le même effet que le nouvel opérateur pour créer une instance de COM la classe. Vous devez utiliser le nouvel opérateur plutôt que cette fonction.

N'utiliser pas <code>com_load()</code> , utilisez la syntaxe OO à la place
<pre><?php // fait cela : \$objj = new COM(\$module); // au lieu de cela : \$objj = com_load(\$module); ?></pre>
Note
Cette fonction n'existe pas en PHP 5 ; utilisez <u>COM</u> à la place.

8.12.22 `com_message_pump()` : Traite un message COM en attendant jusqu'à `timeoutms` millisecondes

bool **com_message_pump** (int *timeoutms*)

com_message_pump traite un message COM en attendant jusqu'à `timeoutms` millisecondes, ou bien en attendant qu'un message arrive dans la file d'attente. Si un ou plusieurs messages arrivent avant le timeout, ils seront dispatchés et la fonction retournera TRUE . Si le timeout survient et qu'aucun message n'est traité, la valeur retournée sera FALSE . Si vous ne spécifiez pas de valeur pour le paramètre `timeoutms` , alors il vaudra 0. Une valeur à 0 signifie que les messages seront traités immédiatement ; s'il y a des messages dans la file d'attente, ils seront dispatchés aussitôt ; s'il n'y a aucun message dans la file d'attente, la fonction retournera FALSE immédiatement sans attendre.

Le but de cette fonction est d'acheminer les appels COM entre les éléments et de gérer les différentes synchronisations. Cela permet à votre script d'attendre efficacement les événements à déclencher, tout en gérant d'autres événements ou bien d'exécuter d'autres scripts en arrière plan. Vous devriez l'utiliser dans une boucle, comme dans l'exemple de la fonction com_event_sink , jusqu'à ce que vous ayez fini d'utiliser les objets COM liés à des événements.

8.12.23 `com_print_typeinfo()` : Affiche une définition de classe PHP pour une interface répartissable

bool **com_print_typeinfo** (object *comobject* , string *dispinterface* , bool *wantsink*)

com_print_typeinfo aide à générer un squelette de classe pour l'utiliser comme évier d'événement. Vous pouvez également l'utiliser pour générer une sauvegarde de n'importe quel objet COM, à condition que il supporte assez d'interfaces d'introspection et que vous sachiez le nom de l'interface que vous désirez afficher.

comobject doit être soit une instance d'un objet COM, soit le nom d'une bibliothèque type (qui doit être résolu en accord des règles définies dans la fonction com_load_typelib). *dispinterface* est le nom d'une interface descendante IDispatch que vous souhaitez afficher. Si le paramètre *wantsink*

vaut TRUE , l'interface d'évier correspondante sera afficher à la place.

Voir aussi [com_event_sink](#) et [com_load_typelib](#) .

8.12.24 [com_propget\(\)](#) : Alias de [com_get](#)

Cette fonction est un alias de : [com_get](#) .

Note

Cette fonction n'existe pas en PHP 5 ; à la place, vous devriez utiliser la syntaxe régulière et plus naturelle OO pour accéder aux propriétés ou appeler les méthodes.

8.12.25 [com_propput\(\)](#) : Alias de [com_set](#)

Cette fonction est un alias de : [com_set](#) .

Note

Cette fonction n'existe pas en PHP 5 ; à la place, vous devriez utiliser la syntaxe régulière et plus naturelle OO pour accéder aux propriétés ou appeler les méthodes.

8.12.26 [com_propset\(\)](#) : Alias de [com_set](#)

Cette fonction est un alias de : [com_set](#) .

Note

Cette fonction n'existe pas en PHP 5 ; à la place, vous devriez utiliser la syntaxe régulière et plus naturelle OO pour accéder aux propriétés ou appeler les méthodes.

8.12.27 [com_release\(\)](#) : Décrémente le compteur de références

void **com_release** (void)

[com_release](#) décrémente le compteur de références.

Attention

Vous ne devriez jamais avoir besoin d'utiliser cette fonction.
--

Note

Cette fonction a été supprimée en PHP 5.
--

8.12.28 [com_set\(\)](#) : Modifie une propriété d'un composant COM

void **com_set** (resource com_object , string property , mixed value)

[com_set](#) remplace la valeur de la propriété property du composante COM com_object par value .
[com_set](#) retourne la nouvelle valeur en cas de succès, et FALSE sinon.

N'utiliser pas com_set() , utilisez la syntaxe OO à la place
--

<pre><?php // faites ceci : \$obj->property = \$value;</pre>
--

```
// à la place de cela :
com_set($obj, 'property', $value);
?>
```

Note

Cette fonction n'existe pas en PHP 5 ; à la place, vous devriez utiliser la syntaxe régulière et plus naturelle OO pour accéder aux propriétés ou appeler les méthodes.

8.12.29 variant_abs() : Retourne la valeur absolue d'un variant

mixed **variant_abs** (mixed val)

Retourne la valeur absolue de val .

Voir aussi [abs](#) .

Note

Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de [VARIANT](#) la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.

Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple variant_add() en PHP correspond à VarAdd() dans la documentation MSDN.

8.12.30 variant_add() : "Ajoute" deux valeurs de variants et retourne le résultat

mixed **variant_add** (mixed left , mixed right)

Ajoute left à right selon les règles suivantes (prises de la bibliothèque MSDN), qui correspondent à celles de Visual Basic :

Si	Alors
Les deux expressions sont des chaînes	Concaténation
Une expression est de type chaîne et l'autre est un caractère	Addition
Une expression est numérique est l'autre est une chaîne	Addition
Les deux exxpressions sont numériques	Addition
Une expression est NULL	NULL est retourné
Les deux expressions sont vides	Le sous-type entier est retourné

Voir aussi [variant_sub](#) .

Note

Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de

VARIANT la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.

Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple `variant_add()` en PHP correspond à `VarAdd()` dans la documentation MSDN.

8.12.31 `variant_and()` : Effectue un ET entre deux variants et retourne le résultat

mixed `variant_and` (mixed left , mixed right)

Effectue un ET sur les bits, selon la table de vérité suivante; notez que cela est différent du ET normal.

Si left est	Si right est	alors le résultat est
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
TRUE	NULL	NULL
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE
FALSE	NULL	FALSE
NULL	TRUE	NULL
NULL	FALSE	FALSE
NULL	NULL	NULL

Voir aussi `variant_or` .

Note

Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de VARIANT la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.

Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple `variant_add()` en PHP correspond à `VarAdd()` dans la documentation MSDN.

8.12.32 `variant_cast()` : Convertit un variant en un nouvel objet variant de type différent

variant `variant_cast` (variant variant , int type)

Cette fonction crée une copie de variant et performe alors un transtypage pour forcer la copie à avoir le type donné par type . type doit être l'une des constantes VT_XXX .

Cette fonction est en fait la fonction `VariantChangeType()` de la bibliothèque COM; consultez MSDN pour plus d'informations.

8.12.30 `variant_add()` : "Ajoute" deux valeurs de variants et retourne le résultat

Voir aussi [variant_set_type](#) .

8.12.33 `variant_cat()` : Concatène deux valeurs variantes ensemble et retourne le résultat

mixed `variant_cat` (mixed left , mixed right)

Concatène left avec right et retourne le résultat.

Voir aussi [Opérateurs de chaînes](#) pour l'opérateur de concaténation des chaînes; cette fonction est équivalente à `$left . $right` .

Note
<p>Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de VARIANT la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.</p> <p>Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple <code>variant_add()</code> en PHP correspond à <code>VarAdd()</code> dans la documentation MSDN.</p>

8.12.34 `variant_cmp()` : Compare deux variants

int `variant_cmp` (mixed left , mixed right , int lcid , int flags)

Compare left avec right et retourne une des valeurs suivantes :

Valeur	Signification
<code>VARCMP_LT</code>	left est plus petit que right
<code>VARCMP_EQ</code>	left est égal à right
<code>VARCMP_GT</code>	left est plus grand que right
<code>VARCMP_NULL</code>	left , right ou les deux sont NULL

Cette fonction ne comparera que des valeurs scalaires, pas de tableaux ni d'enregistrements variants.

lcid est un Locale Identifier valide à utiliser lors des comparaisons des chaînes (cela affecte la collation de la chaîne). flags peut être une ou plusieurs des valeurs suivantes, jointes avec des OR, et affecte les comparaisons de chaînes :

Valeur	Signification
<code>NORM_IGNORECASE</code>	Compare avec sensibilité à la casse
<code>NORM_IGNORENONSPACE</code>	Ignore les caractères non-espaceants
<code>NORM_IGNORESYMBOLS</code>	Ignore les symboles
<code>NORM_IGNOREWIDTH</code>	Ignore la taille de la chaîne
<code>NORM_IGNOREKANATYPE</code>	Ignore le type de Kana
<code>NORM_IGNOREKASHIDA</code>	Ignore les caractères arabes kashida

<p>Note</p> <p>Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de <u>VARIANT</u> la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.</p> <p>Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple <code>variant_add()</code> en PHP correspond à <code>VarAdd()</code> dans la documentation MSDN.</p>

8.12.35 `variant_date_from_timestamp()` : Retourne une représentation date en variant d'un timestamp unix

variant `variant_date_from_timestamp` (int timestamp)

Convertit timestamp depuis un timestamp unix vers un type variant VT_DATE . Cela permet une interopabilité plus facile des parties unix-iennes de PHP et COM.

Voir aussi `variant_date_to_timestamp` pour l'opération inverse, `mktime` et `time` .

8.12.36 `variant_date_to_timestamp()` : Convertit une valeur date/temps variante en un timestamp unix

int `variant_date_to_timestamp` (variant variant)

Convertit variant d'une valeur VT_DATE (ou similaire) en un timestamp unix. Cela permet une interopabilité plus facile entre les parties unix de PHP et COM.

Voir aussi `variant_date_from_timestamp` pour l'inverse de cette opération `date` et `strtotime` .

8.12.37 `variant_div()` : Retourne le résultat de la division de deux variants

mixed `variant_div` (mixed left , mixed right)

Divise left par right et retourne le résultat, qui applique les règles suivantes :

Si	Alors
Les deux expressions sont des chaînes, dates, caractères, booléens	Double est retourné
Une expression est une chaîne et l'autre est un caractère	Division et double retournés
Une expression est numérique et l'autre est une chaîne	Division et double retournés
Les deux expressions sont numériques	Division et double retournés
Une des expressions est NULL	NULL est retourné

right est vide et left est tout mais non vide	Un code com_exception DISP_E_DIVBYZERO est lancé
left est vide et right est tout mais non vide.	0 en tant que double est retourné
Les deux expressions sont vides	Un code com_exception DISP_E_OVERFLOW est lancé

Note

Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de VARIANT la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.

Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple variant_add() en PHP correspond à VarAdd() dans la documentation MSDN.

8.12.38 variant_eqv() : Performe une équivalence de bits de deux variants

mixed **variant_eqv** (mixed left , mixed right)

Si chaque bit dans left est égal au bit correspondant dans right alors la fonction retourne TRUE , sinon elle retourne FALSE .

Note

Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de VARIANT la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.

Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple variant_add() en PHP correspond à VarAdd() dans la documentation MSDN.

8.12.39 variant_fix() : Retourne la portion entière d'un variant

mixed **variant_fix** (mixed variant)

Si variant est négatif, le premier entier négatif plus grand ou égal au variant est retourné, sinon cette fonction retourne la portion entière de la valeur de variant .

Attention

Cette documentation est basée sur la documentation MSDN; il apparaît que cette fonction est la même que variant_int , ou alors, il y'a une erreur dans la documentation de MSDN.

Voir aussi variant_int , variant_round , floor , ceil , round .

Note

8.12.37 variant_div() : Retourne le résultat de la division de deux variants

Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de VARIANT la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.

Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple `variant_add()` en PHP correspond à `VarAdd()` dans la documentation MSDN.

8.12.40 `variant_get_type()` : Retourne le type d'un objet variant

int `variant_get_type` (variant variant)

`variant_get_type` retourne une valeur entière qui indique le type de variant , qui peut être une instance de COM , DOTNET ou de VARIANT . La valeur de retour peut être comparée à l'une des constantes VT_XXX .

La valeur de retour pour les objets COM et DOTNET sera la plupart du temps VT_DISPATCH ; la seule raison qui fait que cette fonction marche avec ces classes est que COM et DOTNET sont de descendants de VARIANT.

Avant PHP 5, vous pouvez obtenir cette information en lisant une fausse propriété type . Voyez la classe VARIANT pour plus d'informations.

8.12.41 `variant_idiv()` : Convertit des variants en valeurs entières, et performe alors une division

mixed `variant_idiv` (mixed left , mixed right)

Convertit left et right en valeurs entières, et performe alors une division entière selon les règles suivantes :

Si	Alors
Les deux expressions sont des chaînes, dates, caractères, booléens	Division et entier retournés
Une expression est une chaîne et l'autre est un caractère	Division
Une expression est numérique et l'autre est une chaîne	Division
Les deux expressions sont numériques	Division
Une des expressions est NULL	NULL est retourné
Les deux expressions sont vides	Un code <code>com_exception</code> <code>DISP_E_DIVBYZERO</code> est lancé

Voir aussi `variant_div` .

Note

Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une

instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de VARIANT la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.

Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple `variant_add()` en PHP correspond à `VarAdd()` dans la documentation MSDN.

8.12.42 `variant_imp()` : Performe une implication sur les bits de deux variants

mixed **variant_imp** (mixed left , mixed right)

Performe une opération d'implication sur les bits, selon la table de vérités suivante :

Si left est	Si right est	alors le résultat est
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	NULL	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	TRUE
FALSE	NULL	TRUE
NULL	TRUE	TRUE
NULL	FALSE	NULL
NULL	NULL	NULL

Note

Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de VARIANT la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.

Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple `variant_add()` en PHP correspond à `VarAdd()` dans la documentation MSDN.

8.12.43 `variant_int()` : Retourne la partie entière d'un variant

mixed **variant_int** (mixed variant)

Si variant est négatif, le premier entier négatif supérieur ou égal à ce variant est retourné, sinon cette fonction retourne la partie entière de la valeur de variant .

Note

Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une

8.12.41 `variant_idiv()` : Convertit des variants en valeurs entières, etperforme alors une division 395

instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de VARIANT la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.

Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple `variant_add()` en PHP correspond à `VarAdd()` dans la documentation MSDN.

Voir aussi [variant_fix](#) , [variant_round](#) , [floor](#) , [ceil](#) et [round](#) .

8.12.44 `variant_mod()` : Divise deux variantes et retourne le reste

mixed `variant_mod` (mixed left , mixed right)

Divise left par right et retourne le reste.

Note
<p>Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de <u>VARIANT</u> la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.</p> <p>Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple <code>variant_add()</code> en PHP correspond à <code>VarAdd()</code> dans la documentation MSDN.</p>

Voir aussi [variant_div](#) et [variant_idiv](#) .

8.12.45 `variant_mul()` : Multiplie les valeurs de deux variants et retourne le résultat

mixed `variant_mul` (mixed left , mixed right)

Multiplie left par right et retourne le résultat selon les règles suivantes :

Si	Alors
Les deux expressions sont du type chaîne, date, caractère ou booléen	Multiplication
Une expression est une chaîne et l'autre est un caractère	Multiplication
Une expression est numérique et l'autre est une chaîne	Multiplication
Les deux expressions sont numériques	Multiplication
Une des expressions est NULL	NULL est retourné
Les deux expressions sont vides	Une chaîne vide est retournée

Les valeurs booléennes sont converties de -1 à FALSE et de 0 à TRUE .

Voir aussi [variant_div](#) , [variant_idiv](#) .

Note

Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de VARIANT la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.

Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple `variant_add()` en PHP correspond à `VarAdd()` dans la documentation MSDN.

8.12.46 variant_neg() : Effectue une négation logique sur un variant

mixed **variant_neg** (mixed variant)

Effectue une négation logique de variant et retourne le résultat.

Note

Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de VARIANT la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.

Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple `variant_add()` en PHP correspond à `VarAdd()` dans la documentation MSDN.

8.12.47 variant_not() : Performe une négation sur les bits sur un variant

mixed **variant_not** (mixed variant)

Performe une négation sur les bits sur le paramètre variant et retourne le résultat. Si variant est NULL, le résultat sera aussi NULL.

Note

Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de VARIANT la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.

Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple `variant_add()` en PHP correspond à `VarAdd()` dans la documentation MSDN.

8.12.48 `variant_or()` : Performe une disjonction logique sur deux variants

mixed `variant_or` (mixed left , mixed right)

Effectue une opération sur les bits de types OR, selon la table de vérité suivante; notez que ceci est différent de l'expression OR normale.

Si left est	Si right est	alors le résultat est
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	NULL	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE
FALSE	NULL	NULL
NULL	TRUE	TRUE
NULL	FALSE	NULL
NULL	NULL	NULL

Note

Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de [VARIANT](#) la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.

Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple `variant_add()` en PHP correspond à `VarAdd()` dans la documentation MSDN.

Voir aussi [variant_and](#) et [variant_xor](#) .

8.12.49 `variant_pow()` : Retourne le résultat de la fonction puissance avec deux variants

mixed `variant_pow` (mixed left , mixed right)

Retourne left à la puissance right .

Voir aussi [pow](#) .

Note

Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de [VARIANT](#) la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.

Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple `variant_add()` en PHP correspond à `VarAdd()` dans la documentation MSDN.

8.12.50 `variant_round()` : Arrondit le variant au nombre spécifié de décimales

mixed `variant_round` (mixed `variant` , int `decimals`)

Retourne la valeur de variant arrondie à `decimals` décimales.

Voir aussi [round](#) .

Note

Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de [VARIANT](#) la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.

Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple `variant_add()` en PHP correspond à `VarAdd()` dans la documentation MSDN.

8.12.51 `variant_set_type()` : Convertit un variant en un autre type "sur-place"

void `variant_set_type` (variant `variant` , int `type`)

Cette fonction est similaire [variant_cast](#) à part que le variant est modifié "sur-place"; aucun nouveau variant n'est créé. Les paramètres de cette fonction ont la même signification que ceux de [variant_cast](#) .

Voir aussi [variant_cast](#) .

8.12.52 `variant_set()` : Assigne une nouvelle valeur pour un objet variant

void `variant_set` (variant `variant` , mixed `value`)

Convertit `value` en un variant et l'assigne à l'objet `variant` . Aucun nouvel objet n'est créé et l'ancienne valeur de `variant` est libérée.

8.12.53 `variant_sub()` : Soustrait la valeur du variant de droite de la valeur de celui de gauche et retourne le résultat

mixed `variant_sub` (mixed `left` , mixed `right`)

Soustrait right de left en utilisant les règles suivantes :

Si	Alors
Les deux expressions sont des chaînes	Soustraction
Une expression est une chaîne et l'autre est un caractère	Soustraction
Une expression est numérique et l'autre est une chaîne	Soustraction.
Les deux expressions sont numériques	Soustraction
Une des expressions est NULL	NULL est retourné
Les deux expressions sont vides	Une chaîne vide est retournée

Voir aussi [variant_add](#) .

Note
<p>Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de VARIANT la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.</p> <p>Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple <code>variant_add()</code> en PHP correspond à <code>VarAdd()</code> dans la documentation MSDN.</p>

8.12.54 `variant_xor()` : Performe une exclusion logique sur deux variants

mixed `variant_xor` (mixed left , mixed right)

Performe une exclusion logique selon la table des vérités suivante :

Si left est	Si right est	lors le résultat est
TRUE	TRUE	FALSE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE
NULL	NULL	NULL

Note
<p>Tout comme les fonctions arithmétiques, les paramètres de cette fonction peuvent être soit un type natif de PHP (entier, chaîne de caractères, nombre à virgule flottante, booléen ou NULL), ou une instance d'une classe COM, VARIANT ou DOTNET. Les types natifs de PHP seront convertis en classe VARIANT en utilisant les mêmes rouages que ceux trouvés dans le constructeur de VARIANT la classe. Les objets COM et DOTNET auront les valeurs de leurs propriétés par défaut et seront utilisés dans les valeurs des VARIANT.</p> <p>Les fonctions arithmétiques VARIANT sont transposés sur des fonctions nommées de la même façon dans la bibliothèque COM ; pour plus d'informations sur ces fonctions, consultez la bibliothèque MSDN. Les fonctions PHP sont nommées légèrement différemment ; par exemple <code>variant_add()</code> en PHP correspond à <code>VarAdd()</code> dans la documentation MSDN.</p>

Voir aussi [variant_and](#) et [variant_or](#) .

8.13 Fonctions ClibPDF

8.13.1 Introduction

L'extension ClibPDF vous permet de créer des documents PDF avec PHP. Les fonctionnalités et API de ClibPDF sont très similaires à [PDFlib](#) . Cette documentation devrait être lue avec le manuel ClibPDF sous la main, car il est beaucoup plus détaillé.

Beaucoup de fonctions natives de ClibPDF et du module PHP, de même que celles de [PDFlib](#) ont le même nom. Toutes les fonctions, hormis [cpdf_open](#) utilisent un pointeur sur un document comme premier paramètre.

Actuellement, ce pointeur n'est pas utilisé en interne, car ClibPDF ne supporte pas la création de plusieurs documents PDF simultanément. En fait, il ne vaut mieux pas l'envisager, car les résultats sont aléatoires. Je ne veux même pas imaginer les problèmes qui pourraient se poser avec les environnements multi-tâches. Selon l'auteur de ClibPDF, cette situation va changer dans les prochaines versions (lorsque cette documentation a été écrite, c'était la version 1.10). Si vous avez besoin de cette fonctionnalité, utilisez [pdflib](#).

Une caractéristique pratique de ClibPDF (et aussi de [PDFlib](#)) est celle de créer le document PDF en mémoire, sans fichiers temporaires. ClibPDF permet aussi de passer les coordonnées avec une unité prédéfinie (ce qui peut être simulé avec [pdf_translate](#) de la bibliothèque [PDFlib](#)).

Un autre atout de ClibPDF est que chaque page peut être modifiée à tout moment même si une nouvelle page a été ouverte. La fonction [cpdf_set_current_page](#) vous permet de quitter temporairement une page et d'en modifier une autre.

La plupart des fonctions sont très simples d'emploi. Le plus difficile est probablement de créer un document PDF simple. L'exemple suivant devrait vous aider à démarrer. La page contient du texte qui utilise la police "Times-Roman" en taille 30, outlined. Le texte est souligné.

Note
Cette extension a été déplacée dans le module PECL et ne sera plus intégrée dans PHP à partir de PHP 5.1.0.
Note
Si vous êtes intéressés par des alternatives gratuites pour générer des PDF, sans passer par des bibliothèques PDF, voyez cette entrée de la FAQ .

8.13.2 Pré-requis

Pour utiliser les fonctions ClibPDF vous devez installer la librairie ClibPDF. Elle est disponible au téléchargement sur le site de [FastIO](#) , mais elle impose l'achat d'une license pour utilisation commerciale. PHP requiert que vous utilisiez ClibPDF >= 2.

8.13.3 Installation

Pour pouvoir utiliser ces fonctions, vous devez compiler PHP avec l'option `--with-cpdflib[=DIR]` . DIR est le dossier d'installation de CPDFLib et, par défaut, il vaut /usr . De plus, vous pouvez spécifier les bibliothèques TIFF et JPEG que vous utilisez. Pour cela, ajoutez les options de configuration `--with-jpeg-dir[=DIR]` et `--with-tiff-dir[=DIR]` .

8.13.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.13.5 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

```

CPDF_PM_NONE (entier)
CPDF_PM_OUTLINES (entier)
CPDF_PM_THUMBS (entier)
CPDF_PM_FULLSCREEN (entier)
CPDF_PL_SINGLE (entier)
CPDF_PL_1COLUMN (entier)
CPDF_PL_2LCOLUMN (entier)
CPDF_PL_2RCOLUMN (entier)

```

8.13.6 Exemples

Exemple simple ClibPDF

```

<?php
$cpdf = cpdf_open(0);
cpdf_page_init($cpdf, 1, 0, 595, 842, 1.0);
cpdf_add_outline($cpdf, 0, 0, 0, 1, "Page 1");
cpdf_begin_text($cpdf);
cpdf_set_font($cpdf, "Times-Roman", 30, "WinAnsiEncoding");
cpdf_set_text_rendering($cpdf, 1);
cpdf_text($cpdf, "Times Roman outlined", 50, 50);
cpdf_end_text($cpdf);
cpdf_moveto($cpdf, 50, 50);
cpdf_lineto($cpdf, 740, 330);
cpdf_stroke($cpdf);
cpdf_finalize_page($cpdf, 1);
cpdf_finalize($cpdf);
Header("Content-type: application/pdf");
cpdf_output_buffer($cpdf);
cpdf_close($cpdf);
?>

```

La distribution pdflib contient un exemple plus complet, qui crée des séries de pages avec une horloge. Voici cet exemple converti en script PHP qui utilise l'extension ClibPDF :

Exemple pdfclose de la distribution pdflib 2.0

```

<?php
$radius = 200;
$margin = 20;
$pagecount = 40;

$pdf = cpdf_open(0);
cpdf_set_creator($pdf, "pdf_clock.php");
cpdf_set_title($pdf, "Analog Clock");

while ($pagecount-- > 0) {
    cpdf_page_init($pdf, $pagecount+1, 0, 2 * ($radius + $margin), 2 * ($radius + $margin), 1.0);
    cpdf_set_page_animation($pdf, 4, 0.5, 0, 0, 0); /* wipe */
}

```

```

cpdf_translate($pdf, $radius + $margin, $radius + $margin);
cpdf_save($pdf);
cpdf_setrgbcolor($pdf, 0.0, 0.0, 1.0);

/* minute strokes */
cpdf_setlinewidth($pdf, 2.0);
for ($alpha = 0; $alpha < 360; $alpha += 6)
{
    cpdf_rotate($pdf, 6.0);
    cpdf_moveto($pdf, $radius, 0.0);
    cpdf_lineto($pdf, $radius-$margin/3, 0.0);
    cpdf_stroke($pdf);
}

cpdf_restore($pdf);
cpdf_save($pdf);

/* 5 minute strokes */
cpdf_setlinewidth($pdf, 3.0);
for ($alpha = 0; $alpha < 360; $alpha += 30)
{
    cpdf_rotate($pdf, 30.0);
    cpdf_moveto($pdf, $radius, 0.0);
    cpdf_lineto($pdf, $radius-$margin, 0.0);
    cpdf_stroke($pdf);
}

$time = getdate();

/* draw hour hand */
cpdf_save($pdf);
cpdf_rotate($pdf, -((($time['minutes']/60.0) + $time['hours'] - 3.0) * 30.0));
cpdf_moveto($pdf, -$radius/10, -$radius/20);
cpdf_lineto($pdf, $radius/2, 0.0);
cpdf_lineto($pdf, -$radius/10, $radius/20);
cpdf_closepath($pdf);
cpdf_fill($pdf);
cpdf_restore($pdf);

/* draw minute hand */
cpdf_save($pdf);
cpdf_rotate($pdf, -((($time['seconds']/60.0) + $time['minutes'] - 15.0) * 6.0));
cpdf_moveto($pdf, -$radius/10, -$radius/20);
cpdf_lineto($pdf, $radius * 0.8, 0.0);
cpdf_lineto($pdf, -$radius/10, $radius/20);
cpdf_closepath($pdf);
cpdf_fill($pdf);
cpdf_restore($pdf);

/* draw second hand */
cpdf_setrgbcolor($pdf, 1.0, 0.0, 0.0);
cpdf_setlinewidth($pdf, 2);
cpdf_save($pdf);
cpdf_rotate($pdf, -((($time['seconds'] - 15.0) * 6.0));
cpdf_moveto($pdf, -$radius/5, 0.0);
cpdf_lineto($pdf, $radius, 0.0);
cpdf_stroke($pdf);
cpdf_restore($pdf);

/* draw little circle at center */
cpdf_circle($pdf, 0, 0, $radius/30);
cpdf_fill($pdf);

cpdf_restore($pdf);

cpdf_finalize_page($pdf, $pagecount+1);

```

```
}  
cpdf_finalize($pdf);  
header("Content-type: application/pdf");  
cpdf_output_buffer($pdf);  
cpdf_close($pdf);  
?>
```

8.13.7 Voir aussi

Voir aussi la documentation de [PDFlib](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [Voir aussi](#)
- [cpdf_add_annotation](#)
- [cpdf_add_outline](#)
- [cpdf_arc](#)
- [cpdf_begin_text](#)
- [cpdf_circle](#)
- [cpdf_clip](#)
- [cpdf_close](#)
- [cpdf_closepath_fill_stroke](#)
- [cpdf_closepath_stroke](#)
- [cpdf_closepath](#)
- [cpdf_continue_text](#)
- [cpdf_curveto](#)
- [cpdf_end_text](#)
- [cpdf_fill_stroke](#)
- [cpdf_fill](#)
- [cpdf_finalize_page](#)
- [cpdf_finalize](#)
- [cpdf_global_set_document_limits](#)
- [cpdf_import_jpeg](#)
- [cpdf_lineto](#)
- [cpdf_moveto](#)
- [cpdf_newpath](#)
- [cpdf_open](#)
- [cpdf_output_buffer](#)
- [cpdf_page_init](#)
- [cpdf_place_inline_image](#)
- [cpdf_rect](#)
- [cpdf_restore](#)
- [cpdf_rlineto](#)
- [cpdf_rmoveto](#)
- [cpdf_rotate_text](#)
- [cpdf_rotate](#)
- [cpdf_save_to_file](#)
- [cpdf_save](#)
- [cpdf_scale](#)

- [cpdf_set_action_url](#)
- [cpdf_set_char_spacing](#)
- [cpdf_set_creator](#)
- [cpdf_set_current_page](#)
- [cpdf_set_font_directories](#)
- [cpdf_set_font_map_file](#)
- [cpdf_set_font](#)
- [cpdf_set_horiz_scaling](#)
- [cpdf_set_keywords](#)
- [cpdf_set_leading](#)
- [cpdf_set_page_animation](#)
- [cpdf_set_subject](#)
- [cpdf_set_text_matrix](#)
- [cpdf_set_text_pos](#)
- [cpdf_set_text_rendering](#)
- [cpdf_set_text_rise](#)
- [cpdf_set_title](#)
- [cpdf_set_viewer_preferences](#)
- [cpdf_set_word_spacing](#)
- [cpdf_setdash](#)
- [cpdf_setflat](#)
- [cpdf_setgray_fill](#)
- [cpdf_setgray_stroke](#)
- [cpdf_setgray](#)
- [cpdf_setlinecap](#)
- [cpdf_setlinejoin](#)
- [cpdf_setlinewidth](#)
- [cpdf_setmiterlimit](#)
- [cpdf_setrgbcolor_fill](#)
- [cpdf_setrgbcolor_stroke](#)
- [cpdf_setrgbcolor](#)
- [cpdf_show_xy](#)
- [cpdf_show](#)
- [cpdf_stringwidth](#)
- [cpdf_stroke](#)
- [cpdf_text](#)
- [cpdf_translate](#)

8.13.9 [cpdf_add_outline\(\)](#) : Ajoute un signet à la page courante

int [cpdf_add_outline](#) (int pdf_document , int lastoutline , int sublevel , int open , int pagenr , string text)

[cpdf_add_outline](#) ajoute un signet avec le texte text qui pointe sur la page courante.

Ajouter une mise en relief
<pre> <?php \$cpdf = cpdf_open(0); cpdf_page_init(\$cpdf, 1, 0, 595, 842); cpdf_add_outline(\$cpdf, 0, 0, 0, 1, "Page 1"); // ... // quelques dessins // ... cpdf_finalize(\$cpdf); header("Content-type: application/pdf"); </pre>

```

cpdf_output_buffer($cpdf);
cpdf_close($cpdf);
?>

```

8.13.10 `cpdf_arc()` : Dessine un arc de cercle

bool **cpdf_arc** (int pdf_document , float x_coor , float y_coor , float radius , float start , float end , int mode)

`cpdf_arc` dessine un arc de cercle dont le centre est au point (x_coor , y_coor) et l'angle est radius , commençant à l'angle start et finissant à l'angle end . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

Le paramètre optionnel mode détermine la longueur de l'unité. S'il prend la valeur 0 (ou s'il est omis), c'est la valeur par défaut définie pour la page qui est utilisée. Sinon, les coordonnées sont mesurées en points PostScript sans se préoccuper de l'unité courante.

Voir aussi [cpdf_circle](#) .

8.13.11 `cpdf_begin_text()` : Démarre une section de texte

bool **cpdf_begin_text** (int pdf_document)

`cpdf_begin_text` démarre une section de texte. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.. La section doit être terminée avec [cpdf_end_text](#) .

Affichage de texte avec `cpdf_begin_text`

```

<?php
cpdf_begin_text($pdf);
cpdf_set_font($pdf, 16, "Helvetica", "WinAnsiEncoding");
cpdf_text($pdf, 100, 100, "Some text");
cpdf_end_text($pdf)
?>

```

Voir aussi [cpdf_end_text](#) .

8.13.12 `cpdf_circle()` : Dessine un cercle

bool **cpdf_circle** (int pdf_document , float x_coor , float y_coor , float radius , int mode)

`cpdf_circle` dessine un cercle de centre (x_coor , y_coor) et de rayon radius . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

Le paramètre optionnel mode détermine la longueur de l'unité. S'il prend la valeur 0 (ou s'il est omis), c'est la valeur par défaut définie pour la page qui est utilisée. Sinon, les coordonnées sont mesurées en points PostScript sans se préoccuper de l'unité courante.

Voir aussi [cpdf_arc](#) .

8.13.13 `cpdf_clip()` : Aligne les dessins sur le chemin courant

`bool cpdf_clip (int pdf_document)`

`cpdf_clip` aligne les dessins sur le chemin courant. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

8.13.14 `cpdf_close()` : Ferme un fichier PDF

`bool cpdf_close (int pdf_document)`

`cpdf_close` ferme un fichier PDF. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.. Ce doit être la dernière fonction appelée, et elle apparaît même après `cpdf_finalize` , `cpdf_output_buffer` et `cpdf_save_to_file` .

Voir aussi `cpdf_open` .

8.13.15 `cpdf_closepath_fill_stroke()` : Remplit le chemin, dessine le bord et ferme le chemin

`bool cpdf_closepath_fill_stroke (int pdf_document)`

`cpdf_closepath_fill_stroke` remplit le chemin, dessine le bord et ferme le chemin. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

Voir aussi `cpdf_closepath` , `cpdf_stroke` , `cpdf_fill` , `cpdf_setgray_fill` , `cpdf_setgray` , `cpdf_setrgbcolor_fill` et `cpdf_setrgbcolor` .

8.13.16 `cpdf_closepath_stroke()` : Ferme le fichier et dessine une ligne le long du chemin

`bool cpdf_closepath_stroke (int pdf_document)`

`cpdf_closepath_stroke` est une combinaison de `cpdf_closepath` et `cpdf_stroke` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

Voir aussi `cpdf_closepath` et `cpdf_stroke` .

8.13.17 `cpdf_closepath()` : Ferme le chemin

`bool cpdf_closepath (int pdf_document)`

`cpdf_closepath` ferme le chemin courant. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

8.13.18 `cpdf_continue_text()` : Affiche le texte à la ligne suivante

`bool cpdf_continue_text (int pdf_document , string text)`

`cpdf_continue_text` imprime le texte `text` à la ligne suivante. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `cpdf_show_xy` , `cpdf_text` , `cpdf_set_leading` et `cpdf_set_text_pos` .

8.13.19 `cpdf_curveto()` : Dessine une courbe

`bool cpdf_curveto (int pdf_document , float x1 , float y1 , float x2 , float y2 , float x3 , float y3 , int mode)`

`cpdf_curveto` dessine une courbe de Bezier, entre le point courant et le point (`x3` , `y3`), en utilisant les points de contrôle (`x1` , `y1`) et (`x2` , `y2`). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

Le paramètre optionnel `mode` détermine la longueur de l'unité. S'il prend la valeur 0 (ou s'il est omis), c'est la valeur par défaut définie pour la page qui est utilisée. Sinon, les coordonnées sont mesurées en points PostScript sans se préoccuper de l'unité courante.

Voir aussi `cpdf_moveto` , `cpdf_rmoveto` , `cpdf_rlineto` et `cpdf_lineto` .

8.13.20 `cpdf_end_text()` : Termine une section de texte

`bool cpdf_end_text (int pdf_document)`

`cpdf_end_text` termine une section de texte, commencée avec `cpdf_begin_text` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

Affichage de texte

```
<?php
cpdf_begin_text($pdf);
cpdf_set_font($pdf, 16, "Helvetica", "WinAnsiEncoding");
cpdf_text($pdf, 100, 100, "Du texte");
cpdf_end_text($pdf)
?>
```

Voir aussi `cpdf_begin_text` .

8.13.21 `cpdf_fill_stroke()` : Remplit le chemin, et dessine le bord

`bool cpdf_fill_stroke (int pdf_document)`

`cpdf_fill_stroke` remplit l'intérieur du chemin avec la couleur courante, et dessine le chemin. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

Voir aussi `cpdf_closepath` , `cpdf_stroke` , `cpdf_fill` , `cpdf_setgray_fill` , `cpdf_setgray` , `cpdf_setrgbcolor_fill` et `cpdf_setrgbcolor` .

8.13.22 `cpdf_fill()` : Remplit le chemin courant

`bool cpdf_fill (int pdf_document)`

`cpdf_fill` remplit l'intérieur du chemin courant avec la couleur courante. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

Voir aussi `cpdf_closepath` , `cpdf_stroke` , `cpdf_setgray_fill` , `cpdf_setgray` , `cpdf_setrgbcolor_fill` et `cpdf_setrgbcolor` .

8.13.23 `cpdf_finalize_page()` : Termine une page

`bool cpdf_finalize_page (int pdf_document , int page_number)`

`cpdf_finalize_page` termine la page numéro `page_number` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

`cpdf_finalize_page` ne sert qu'à libérer de la mémoire. Les pages terminées prennent moins de place, mais ne peuvent plus être modifiées.

Voir aussi `cpdf_page_init` .

8.13.24 `cpdf_finalize()` : Termine un document

`bool cpdf_finalize (int pdf_document)`

`cpdf_finalize` termine un document. Vous devrez quand-même appeler `cpdf_close` après. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

Voir aussi `cpdf_close` .

8.13.25 `cpdf_global_set_document_limits()` : Fixe les limites d'un document PDF

`bool cpdf_global_set_document_limits (int maxpages , int maxfonts , int maximages , int maxannotations , int maxobjects)`

`cpdf_global_set_document_limits` permet de fixer plusieurs limites au document PDF. Elle doit être appelée avant `cpdf_open` pour être effective. Elle fixe les limites de tous les documents ouverts par la suite. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `cpdf_open` .

8.13.26 `cpdf_import_jpeg()` : Ouvre une image JPEG

`bool cpdf_import_jpeg (int pdf_document , string file_name , float x_coor , float y_coor , float angle , float width , float height , float x_scale , float y_scale , int gsave , int mode)`

cpdf_import_jpeg ouvre une image JPG, enregistrée dans le fichier `file_name` . Le format de l'image doit être JPEG. L'image est placée dans la page courante, aux coordonnées (`x_coor` , `y_coor`). L'image subira une rotation d'un angle de `angle` degrés. `gsave` doit être non nul pour que cette cpdf_import_jpeg fonctionne correctement.

Le paramètre optionnel `mode` détermine la longueur de l'unité. S'il prend la valeur 0 (ou s'il est omis), c'est la valeur par défaut définie pour la page qui est utilisée. Sinon, les coordonnées sont mesurées en points PostScript sans se préoccuper de l'unité courante.

Voir aussi cpdf_place_inline_image .

8.13.27 cpdf_lineto() : Dessine une ligne

`bool cpdf_lineto (int pdf_document , float x_coor , float y_coor , int mode)`

cpdf_lineto dessine une ligne entre le point courant et le point de coordonnées (`x_coor` , `y_coor`). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre optionnel `mode` détermine la longueur de l'unité. S'il prend la valeur 0 (ou s'il est omis), c'est la valeur par défaut définie pour la page qui est utilisée. Sinon, les coordonnées sont mesurées en points PostScript sans se préoccuper de l'unité courante.

Voir aussi cpdf_moveto , cpdf_rmoveto et cpdf_curveto .

8.13.28 cpdf_moveto() : Fixe le point courant de dessin CPDF

`bool cpdf_moveto (int pdf_document , float x_coor , float y_coor , int mode)`

cpdf_moveto fixe le point courant aux coordonnées (`x_coor` , `y_coor`). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre optionnel `mode` détermine la longueur de l'unité. S'il prend la valeur 0 (ou s'il est omis), c'est la valeur par défaut définie pour la page qui est utilisée. Sinon, les coordonnées sont mesurées en points PostScript sans se préoccuper de l'unité courante.

8.13.29 cpdf_newpath() : Commence un nouveau chemin

`bool cpdf_newpath (int pdf_document)`

cpdf_newpath commence un nouveau chemin dans le document `pdf_document` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.13.30 cpdf_open() : Ouvre un nouveau document PDF

`int cpdf_open (int compression , string filename , array doc_limits)`

cpdf_open ouvre un nouveau document PDF. Le premier paramètre `compression` active ou pas la compression, suivant qu'il vaut 0 ou 1. Le deuxième paramètre optionnel choisit le fichier de destination du document. S'il est omis, le document sera écrit en mémoire et pourra être écrit dans

un fichier avec `cpdf_save_to_file` ou envoyé à l'affichage avec `cpdf_output_buffer` .

Note

La valeur retournée sera nécessaire pour les autres fonctions de ClibPDF comme premier paramètre.

La bibliothèque ClibPDF prend le nom de fichier "-" comme synonyme de stdout. Si PHP est compilé comme un module apache, cela ne fonctionnera pas, car la méthode d'envoi des données de ClibPDF ne fonctionne pas avec Apache. Vous pouvez résoudre ce problème en ne fournissant pas de nom de fichier, et en utilisant la fonction `cpdf_output_buffer` pour afficher le document PDF.

Voir aussi `cpdf_output_buffer` et `cpdf_close` .

8.13.31 `cpdf_output_buffer()` : Affiche le document PDF qui est en mémoire

`bool cpdf_output_buffer (int pdf_document)`

`cpdf_output_buffer` envoie le document PDF `pdf_document` vers la sortie standard (le navigateur sur un serveur web). Le document doit avoir été créé en mémoire, ce qui est le cas si la fonction `cpdf_open` a été appelée sans spécifier de nom de fichier. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `cpdf_open` .

8.13.32 `cpdf_page_init()` : Commence une nouvelle page

`bool cpdf_page_init (int pdf_document , int page_number , int orientation , float height , float width , float unit)`

`cpdf_page_init` commence une nouvelle page de hauteur `height` et de largeur `width` , de numéro de page `page_number` et d'orientation `orientation` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

`orientation` vaut 0 pour le mode portrait et 1 pour le mode paysage. Le dernier paramètre optionnel `unit` configure l'unité du système de coordonnées. Sa valeur doit être exprimée en points postscript par unité. Comme un pouce (2.54 cm) vaut 72 points, une valeur de 72 configure l'unité de mesure d'un pouce. Par défaut, ce paramètre vaut 72.

Voir aussi `cpdf_set_current_page` .

8.13.33 `cpdf_place_inline_image()` : Place une image dans une page

`bool cpdf_place_inline_image (int pdf_document , int image , float x_coor , float y_coor , float angle , float width , float height , int gsave , int mode)`

`cpdf_place_inline_image` place l'image `image` , créée avec les fonctions images de PHP, à la position (`x_coor` , `y_coor`). L'échelle de l'image peut aussi modifiée avec les paramètres `width` et `height` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre optionnel mode détermine la longueur de l'unité. S'il prend la valeur 0 (ou s'il est omis), c'est la valeur par défaut définie pour la page qui est utilisée. Sinon, les coordonnées sont mesurées en points PostScript sans se préoccuper de l'unité courante.

Note

Cette fonction est uniquement disponible si PHP a été compilé avec la bibliothèque GD version 1.3. Lisez les instructions d'installation de l' [extension GD](#) pour plus d'informations.

Voir aussi [cpdf_import_jpeg](#) .

8.13.34 cpdf_rect() : Dessine un rectangle

bool **cpdf_rect** (int pdf_document , float x_coor , float y_coor , float width , float height , int mode)

[cpdf_rect](#) dessine un rectangle dont le coin inférieur gauche de coordonnées est (x_coor , y_coor). La largeur vaut width . La hauteur vaut height . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre optionnel mode détermine la longueur de l'unité. S'il prend la valeur 0 (ou s'il est omis), c'est la valeur par défaut définie pour la page qui est utilisée. Sinon, les coordonnées sont mesurées en points PostScript sans se préoccuper de l'unité courante.

Dessiner un rectangle

```
<?php
$cpdf = cpdf_open(0);
cpdf_page_init($cpdf, 1, 0, 595, 842, 1.0);

// couleur de remplissage (rouge)
cpdf_setrgbcolor($cpdf, 1, 0, 0);

// dessine un rectangle (180 * 100)
cpdf_rect($cpdf, 645, 400, 180, 100);

// colorie le rectangle
cpdf_fill($cpdf);

cpdf_finalize($cpdf);
header("Content-type: application/pdf");
cpdf_output_buffer($cpdf);
cpdf_close($cpdf);

?>
```

8.13.35 cpdf_restore() : Restaure un environnement sauvegardé

bool **cpdf_restore** (int pdf_document)

[cpdf_restore](#) restaure l'environnement sauvé par [cpdf_save](#) . Cette fonction est similaire à la commande postscript grestore. Très pratique quand vous devez faire des translations et rotations sur un objet, mais sans affecter les autres. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Sauver/Restaure l'environnement CPDF

```

<?php
cpdf_save($pdf);
// plein de transformations, translations, ...
cpdf_restore($pdf)
?>

```

Voir aussi [cpdf_save](#) .

8.13.36 cpdf_rlineto() : Dessine une ligne, relativement

bool **cpdf_rlineto** (int pdf_document , float x_coor , float y_coor , int mode)

[cpdf_rlineto](#) dessine une ligne entre le point courant et le point de coordonnées relatives (x_coor , y_coor). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre optionnel mode détermine la longueur de l'unité. S'il prend la valeur 0 (ou s'il est omis), c'est la valeur par défaut définie pour la page qui est utilisée. Sinon, les coordonnées sont mesurées en points PostScript sans se préoccuper de l'unité courante.

Voir aussi [cpdf_moveto](#) , [cpdf_rmoveto](#) et [cpdf_curveto](#) .

8.13.37 cpdf_rmoveto() : Fixe le point courant relativement

bool **cpdf_rmoveto** (int pdf_document , float x_coor , float y_coor , int mode)

[cpdf_rmoveto](#) fixe le point courant relativement aux coordonnées (x_coor , y_coor). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre optionnel mode détermine la longueur de l'unité. S'il prend la valeur 0 (ou s'il est omis), c'est la valeur par défaut définie pour la page qui est utilisée. Sinon, les coordonnées sont mesurées en points PostScript sans se préoccuper de l'unité courante.

Voir aussi [cpdf_moveto](#) .

8.13.38 cpdf_rotate_text() : Configure l'inclinaison d'un texte

bool **cpdf_rotate_text** (int pdfdoc , float angle)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.13.39 cpdf_rotate() : Effectue une rotation

bool **cpdf_rotate** (int pdf_document , float angle)

[cpdf_rotate](#) effectue une rotation d'un angle de angle degrés. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.13.40 `cpdf_save_to_file()` : Écrit un document PDF dans un fichier

`bool cpdf_save_to_file (int pdf_document , string filename)`

`cpdf_save_to_file` écrit un document PDF dans un fichier, s'il a été créé en mémoire. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

`cpdf_save_to_file` n'est pas nécessaire si un nom de fichier a été fourni lors de l'appel à `cpdf_open`.

Voir aussi `cpdf_output_buffer` et `cpdf_open`.

8.13.41 `cpdf_save()` : Sauve l'environnement courant

`bool cpdf_save (int pdf_document)`

`cpdf_save` sauve l'environnement courant. `cpdf_save` est similaire à la commande postscript `gsave`. Très pratique quand vous devez faire des translations et rotations sur un objet, mais sans affecter les autres. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `cpdf_restore`.

8.13.42 `cpdf_scale()` : Modifie l'échelle

`bool cpdf_scale (int pdf_document , float x_scale , float y_scale)`

`cpdf_scale` modifie l'échelle dans les deux directions. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.13.43 `cpdf_set_action_url()` : Paramètre un hyper lien

`bool cpdf_set_action_url (int pdfdoc , float xll , float yll , float xur , float yur , string url , int mode)`

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.13.44 `cpdf_set_char_spacing()` : Fixe l'espacement des caractères

`bool cpdf_set_char_spacing (int pdf_document , float space)`

`cpdf_set_char_spacing` fixe l'espacement des caractères à `space`. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `cpdf_set_word_spacing` et `cpdf_set_leading`.

8.13.45 `cpdf_set_creator()` : Fixe le créateur d'un document PDF

`bool cpdf_set_creator (int pdf_document , string creator)`

`cpdf_set_creator` fixe le créateur d'un document PDF. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `cpdf_set_subject` , `cpdf_set_title` et `cpdf_set_keywords` .

8.13.46 `cpdf_set_current_page()` : Fixe la page courante

bool `cpdf_set_current_page` (int pdf_document , int page_number)

`cpdf_set_current_page` fixe la page courante, où toutes les prochaines opérations auront lieu. On peut changer de page jusqu'à ce qu'une page soit terminée avec `cpdf_finalize_page` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `cpdf_finalize_page` .

8.13.47 `cpdf_set_font_directories()` : Précise les dossiers à fouiller pour rechercher des polices externes

bool `cpdf_set_font_directories` (int pdfdoc , string pfmdir , string pfbdir)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.13.48 `cpdf_set_font_map_file()` : Configure le fichier de remplacement de polices externes

bool `cpdf_set_font_map_file` (int pdfdoc , string filename)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.13.49 `cpdf_set_font()` : Sélectionne la police courante et sa taille

bool `cpdf_set_font` (int pdf_document , string font_name , float size , string encoding)

`cpdf_set_font` sélectionne la police courante, sa taille et son encodage. Actuellement, seules les polices postscript sont supportées. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le dernier paramètre encoding peut prendre les valeurs suivantes : "MacRomanEncoding", "MacExpertEncoding", "WinAnsiEncoding", et " NULL ". " NULL " signifie qu'il faut utiliser l'encodage embarqué.

Reportez-vous à la documentation de ClibPDF, pour plus d'informations, notamment pour l'utilisation des polices asiatiques.

8.13.50 `cpdf_set_horiz_scaling()` : Fixe l'échelle horizontale du texte

`bool cpdf_set_horiz_scaling (int pdf_document , float scale)`

`cpdf_set_horiz_scaling` fixe l'échelle horizontale du texte à `scale %`. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

8.13.51 `cpdf_set_keywords()` : Fixe les mots-clés d'un document PDF

`bool cpdf_set_keywords (int pdf_document , string keywords)`

`cpdf_set_keywords` fixe les mots-clés d'un document PDF. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec..

Voir aussi `cpdf_set_title` , `cpdf_set_creator` et `cpdf_set_subject` .

8.13.52 `cpdf_set_leading()` : Fixe la distance entre deux lignes

`bool cpdf_set_leading (int pdf_document , float distance)`

`cpdf_set_leading` fixe la distance entre deux lignes. Cela servira si le texte est affiché par `cpdf_continue_text` . Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi `cpdf_continue_text` .

8.13.53 `cpdf_set_page_animation()` : Fixe l'animation de la transition entre les pages

`bool cpdf_set_page_animation (int pdf_document , int transition , float duration , float direction , int orientation , int inout)`

`cpdf_set_page_animation` fixe l'animation de la transition entre les pages. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

La valeur du paramètre de transition peut être :

- 0 pour aucune,
- 1 pour deux lignes en travers de l'écran, qui révèlent la prochaine page,
- 2 pour plusieurs lignes en travers de l'écran, qui révèlent la prochaine page,
- 3 pour une boîte qui révèle la prochaine page,
- 4 pour une seule ligne en travers de l'écran, qui révèle la prochaine page,
- 5 pour l'ancienne page qui se dissout
- 6 pour un effet de dissolution d'un angle à l'autre
- 7 pour le remplacement simple (par défaut)

La valeur de duration est le nombre de secondes avant le passage à la page suivante.

8.13.54 `cpdf_set_subject()` : Fixe le sujet d'un document PDF

`bool cpdf_set_subject (int pdf_document , string subject)`

`cpdf_set_subject` fixe le sujet d'un document PDF. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `cpdf_set_title` , `cpdf_set_creator` et `cpdf_set_keywords` .

8.13.55 `cpdf_set_text_matrix()` : Fixe la matrice du texte

`bool cpdf_set_text_matrix (int pdf_document , array matrix)`

`cpdf_set_text_matrix` fixe la matrice du texte qui décrit la transformation appliquée à la police courante. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.13.56 `cpdf_set_text_pos()` : Fixe la position du texte

`bool cpdf_set_text_pos (int pdf_document , float x_coor , float y_coor , int mode)`

`cpdf_set_text_pos` fixe la position du texte pour le prochain appel à `cpdf_show` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre optionnel mode détermine la longueur de l'unité. S'il prend la valeur 0 (ou s'il est omis), c'est la valeur par défaut définie pour la page qui est utilisée. Sinon, les coordonnées sont mesurées en points PostScript sans se préoccuper de l'unité courante.

Voir aussi `cpdf_show` et `cpdf_text` .

8.13.57 `cpdf_set_text_rendering()` : Détermine le rendu du texte

`bool cpdf_set_text_rendering (int pdf_document , int rendermode)`

`cpdf_set_text_rendering` détermine le mode de rendu du texte. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Les valeurs possibles pour rendermode sont : 0=texte plein, 1=texte stroke, 2=texte plein et stroke, 3=invisible, 4=texte plein et ajouté au chemin, 5=texte stroke et ajouté au chemin, 6=texte plein et stroke et ajouté au chemin, 7=ajouté au chemin.

8.13.58 `cpdf_set_text_rise()` : Fixe l'élévation du texte

`bool cpdf_set_text_rise (int pdf_document , float value)`

`cpdf_set_text_rise` fixe l'élévation du texte à valeur unités. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.13.59 `cpdf_set_title()` : Fixe le titre d'un document PDF

`bool cpdf_set_title (int pdf_document , string title)`

`cpdf_set_title` fixe le titre d'un document PDF. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `cpdf_set_subject` , `cpdf_set_creator` et `cpdf_set_keywords` .

8.13.60 `cpdf_set_viewer_preferences()` : Indique comment afficher le document à l'utilisateur

`bool cpdf_set_viewer_preferences (int pdfdoc , array preferences)`

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.13.61 `cpdf_set_word_spacing()` : Fixe l'espacement des mots

`bool cpdf_set_word_spacing (int pdf_document , float space)`

`cpdf_set_word_spacing` fixe l'espacement des mots. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `cpdf_set_char_spacing` et `cpdf_set_leading` .

8.13.62 `cpdf_setdash()` : Fixe le motif de pointillés

`bool cpdf_setdash (int pdf_document , float white , float black)`

`cpdf_setdash` fixe le motif de pointillés à white unités de blanc et black unités de noir. Si les deux sont à 0, une ligne pleine est fixée. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.13.63 `cpdf_setflat()` : Fixe la platitude (flatness)

`bool cpdf_setflat (int pdf_document , float value)`

`cpdf_setflat` fixe la platitude (flatness) à une valeur comprise entre 0 et 100. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

8.13.64 `cpdf_setgray_fill()` : Modifie le niveau de gris comme couleur de remplissage

`bool cpdf_setgray_fill (int pdf_document , float value)`

`cpdf_setgray_fill` remplace le niveau de gris, couleur de remplissage courante, par value . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [cpdf_setrgbcolor_fill](#) .

8.13.65 [cpdf_setgray_stroke\(\)](#) : Choisit un niveau de gris comme couleur de dessin

bool [cpdf_setgray_stroke](#) (int pdf_document , float gray_value)

[cpdf_setgray_stroke](#) remplace le niveau de gris, couleur de dessin courante, par value . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [cpdf_setrgbcolor_stroke](#) .

8.13.66 [cpdf_setgray\(\)](#) : Modifie un niveau de gris comme couleur de dessin et de remplissage

bool [cpdf_setgray](#) (int pdf_document , float gray_value)

[cpdf_setgray_stroke](#) remplace le niveau de gris, couleur de dessin et de remplissage, par gray_value . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [cpdf_setrgbcolor_stroke](#) et [cpdf_setrgbcolor_fill](#) .

8.13.67 [cpdf_setlinecap\(\)](#) : Fixe le paramètre linecap

bool [cpdf_setlinecap](#) (int pdf_document , int value)

[cpdf_setlinecap](#) fixe le paramètre linecap à la valeur value comprise entre 0 et 2. 0 = butt end, 1 = round, 2 = projecting square. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.13.68 [cpdf_setlinejoin\(\)](#) : Fixe le paramètre linejoin

bool [cpdf_setlinejoin](#) (int pdf_document , int value)

[cpdf_setlinejoin](#) fixe le paramètre linejoin à une valeur value , entre 0 et 2. 0 = miter, 1 = round, 2 = bevel. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.13.69 [cpdf_setlinewidth\(\)](#) : Fixe la largeur de ligne

bool [cpdf_setlinewidth](#) (int pdf_document , float width)

[cpdf_setlinewidth](#) fixe la largeur de ligne à la valeur de width . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.13.70 `cpdf_setmiterlimit()` : Fixe le paramètre miter limit

`bool cpdf_setmiterlimit (int pdf_document , float value)`

`cpdf_setmiterlimit` fixe le paramètre "miter limit" à une valeur supérieure ou égale à 1. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

8.13.71 `cpdf_setrgbcolor_fill()` : Choisit une couleur rgb comme couleur de remplissage

`bool cpdf_setrgbcolor_fill (int pdf_document , float red_value , float green_value , float blue_value)`

`cpdf_setrgbcolor_fill` remplace la couleur de remplissage, par la couleur rgb (`red_value` , `green_value` , `blue_value`). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Les valeurs doivent être des nombres à virgule flottante compris entre 0.0 et 1.0 (le noir est (0.0, 0.0, 0.0) et le blanc est (1.0, 1.0, 1.0)).
--

Voir aussi `cpdf_setrgbcolor_stroke` et `cpdf_setrgbcolor` .

8.13.72 `cpdf_setrgbcolor_stroke()` : Choisit une couleur rgb comme couleur de dessin

`bool cpdf_setrgbcolor_stroke (int pdf_document , float red_value , float green_value , float blue_value)`

`cpdf_setrgbcolor_stroke` remplace la couleur de dessin, par la couleur rgb (`red_value` , `green_value` , `blue_value`). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Les valeurs doivent être des nombres à virgule flottante compris entre 0.0 et 1.0 (le noir est (0.0, 0.0, 0.0) et le blanc est (1.0, 1.0, 1.0)).
--

Voir aussi `cpdf_setrgbcolor_fill` et `cpdf_setrgbcolor` .

8.13.73 `cpdf_setrgbcolor()` : Choisit une couleur rgb comme couleur de dessin et de remplissage

`bool cpdf_setrgbcolor (int pdf_document , float red_value , float green_value , float blue_value)`

`cpdf_setrgbcolor_stroke` remplace la couleur de remplissage et de dessin, par la couleur rgb (`red_value` , `green_value` , `blue_value`). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Les valeurs doivent être des nombres à virgule flottante compris entre 0.0 et 1.0 (le noir est (0.0, 0.0, 0.0) et le blanc est (1.0, 1.0, 1.0)).

Voir aussi [cpdf_setrgbcolor_stroke](#) et [cpdf_setrgbcolor_fill](#) .

8.13.74 [cpdf_show_xy\(\)](#) : Affiche un texte à une position

bool **cpdf_show_xy** (int pdf_document , string text , float x_coor , float y_coor , int mode)

[cpdf_show_xy](#) imprime la chaîne text , à la position de coordonnées (x-coor , y-coor). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre optionnel mode détermine la longueur de l'unité. S'il prend la valeur 0 (ou s'il est omis), c'est la valeur par défaut définie pour la page qui est utilisée. Sinon, les coordonnées sont mesurées en points PostScript sans se préoccuper de l'unité courante.

Note

La fonction [cpdf_show_xy](#) est identique à [cpdf_text](#) , sans les options.

Voir aussi [cpdf_text](#) .

8.13.75 [cpdf_show\(\)](#) : Imprime un texte à la position courante

bool **cpdf_show** (int pdf_document , string text)

[cpdf_show](#) imprime la chaîne text à la position courante. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [cpdf_text](#) , [cpdf_begin_text](#) et [cpdf_end_text](#) .

8.13.76 [cpdf_stringwidth\(\)](#) : Retourne la taille de la police courante

float **cpdf_stringwidth** (int pdf_document , string text)

[cpdf_stringwidth](#) retourne la taille de la chaîne text . Une police doit avoir déjà été choisie.

Voir aussi [cpdf_set_font](#) .

8.13.77 [cpdf_stroke\(\)](#) : Dessine une ligne le long du chemin

bool **cpdf_stroke** (int pdf_document)

[cpdf_stroke](#) dessine une ligne le long du chemin courant. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [cpdf_closepath](#) et [cpdf_closepath_stroke](#) .

8.13.78 `cpdf_text()` : Imprime un texte avec des options

bool **cpdf_text** (int pdf_document , string text , float x_coor , float y_coor , int mode , float orientation , int alignmode)

`cpdf_text` imprime le text text à la position de coordonnées (x_coor , y_coor). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre optionnel mode détermine la longueur de l'unité. S'il prend la valeur 0 (ou s'il est omis), c'est la valeur par défaut définie pour la page qui est utilisée. Sinon, les coordonnées sont mesurées en points PostScript sans se préoccuper de l'unité courante.

Le paramètre optionnel orientation est un angle de rotation du texte, en degrés.

Le paramètre optionnel alignmode détermine l'alignement du texte.

Reportez-vous à la documentation de ClibPDF pour les valeurs possibles.

Voir aussi `cpdf_show_xy` .

8.13.79 `cpdf_translate()` : Modifie l'origine du système de coordonnées

bool **cpdf_translate** (int pdf_document , float x_coor , float y_coor)

`cpdf_translate` modifie l'origine du système de coordonnées en plaçant l'origine aux coordonnées (x_coor , y_coor). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre optionnel mode détermine la longueur de l'unité. S'il prend la valeur 0 (ou s'il est omis), c'est la valeur par défaut définie pour la page qui est utilisée. Sinon, les coordonnées sont mesurées en points PostScript sans se préoccuper de l'unité courante.

8.14 Crack

8.14.1 Introduction

Ces fonctions vous permettent d'utiliser la bibliothèque CrackLib, pour tester la robustesse de vos mots de passe. Cette bibliothèque teste la taille, les majuscules et minuscules, et effectue des recherches dans le dictionnaire CrackLib. CrackLib donne aussi des conseils pour renforcer vos mots de passe.

Note
Cette extension a été déplacée dans le module <u>PECL</u> et ne sera plus intégrée dans PHP à partir de PHP 5.0.0.

8.14.2 Pré-requis

Plus d'informations concernant la bibliothèque CrackLib sont disponibles à <http://www.crypticide.com/users/alecm/> .

8.14.3 Installation

Cette extension PECL n'est pas intégrée à PHP. Des informations comme les notes sur les nouvelles versions, les téléchargements, les sources des fichiers, les informations concernant les mainteneurs ainsi qu'un CHANGELOG, peuvent être trouvées ici : <http://pecl.php.net/package/crack> .

En PHP 4, les sources de cette extension PECL peuvent être trouvées dans le dossier ext/ avec les sources de PHP ou sur le lien PECL ci-dessous. Si vous voulez utiliser ces fonctions, vous devez compiler PHP avec le support Crack en utilisant l'option de configuration --with-crack[=DIR] .

Les utilisateurs de Windows doivent activer la bibliothèque php_crack.dll dans le php.ini pour pouvoir utiliser ces fonctions. En PHP 4, cette bibliothèque DLL se trouve dans le dossier extensions/ avec les binaires PHP pour Windows téléchargées. Vous pouvez télécharger ces bibliothèques DLL pour ces extensions PECL depuis la page de téléchargement de PHP sous sur la page <http://snaps.php.net/> .

8.14.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

Nom	Par défaut	Modifiable	Historique
crack.default_dictionary	NULL	PHP_INI_PERDIR	Disponible depuis PHP 4.0.5.

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à Directives du php.ini .

8.14.5 Types de ressources

Cette extension ne définit aucune ressource.

8.14.6 Constantes pré-définies

Cette extension ne définit aucune constante.

8.14.7 Exemples

Cet exemple montre comment ouvrir un dictionnaire CrackLib, tester un mot de passe, lire le diagnostic et refermer le dictionnaire.

Exemple CrackLib
<pre><?php /* Ouverture du dictionnaire CrackLib */ \$dictionary = crack_opendict('/usr/local/lib/pw_dict') or die('Unable to open CrackLib dictionary'); /* Vérification du mot de passe */ \$check = crack_check(\$dictionary, 'gx9A2s0x'); /* Lecture du diagnostic */ \$diag = crack_getlastmessage(); echo \$diag; // 'strong password' /* Fermeture du dictionnaire */ crack_closedict(\$dictionary); ?></pre>
Note
<p>Si crack_check retourne TRUE , crack_getlastmessage retournera le message 'strong password' (mot de passe robuste).</p>

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [crack_check](#)
- [crack_closedict](#)
- [crack_getlastmessage](#)
- [crack_opendict](#)

8.14.9 crack_closedict() : Referme le dictionnaire CrackLib

bool **crack_closedict** (*resource dictionary*)

[crack_closedict](#) ferme le dictionnaire représenté par l'identifiant dictionary .

Attention
<p>Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.</p>

8.14.10 `crack_getlastmessage()` : Retourne le message de diagnostic

string `crack_getlastmessage` (void)

`crack_getlastmessage` retourne le message de diagnostic depuis la dernière vérification.

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

8.14.11 `crack_opendict()` : Ouvre un nouveau dictionnaire CrackLib

resource `crack_opendict` (string dictionary)

`crack_opendict` ouvre le dictionnaire CrackLib dictionary spécifié pour être utilisé avec la fonction `crack_check` .

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

Note

Un seul dictionnaire peut être ouvert actuellement.

8.15 Caractères

8.15.1 Introduction

Ces fonctions vérifient si un caractère ou une chaîne de caractères font partie d'une certaine classe de caractères, en fonction de la configuration locale (voir aussi [setlocale](#)).

Appelées avec un argument de type entier, ces fonctions se comportent exactement comme leur équivalent en langage C, du fichier ctype.h . Cela signifie que si vous passez un entier plus petit que 256, la fonction utilisera la valeur ASCII de cet entier pour voir s'il se trouve dans l'intervalle spécifique (les digitaux sont dans l'intervalle 0x30-0x39). Si le nombre est entre -128 (inclusif) et 0, alors 256 sera ajouté et la vérification sera effectuée sur le nombre résultant.

Appelées avec un argument de type chaîne, elles vérifieront chaque caractère de la chaîne, et ne retourneront TRUE que si chacun satisfait les critères requis. Appelées avec un argument vide, le résultat sera toujours TRUE .

Tout autre type d'argument (autre que chaîne ou entier) génère une erreur, et retourne FALSE immédiatement.

Il est à noter que les fonctions ctype sont toujours préférables par rapport aux expressions rationnelles ainsi que les fonctions équivalentes str_* et is_*. Les fonctions ctype utilisent une bibliothèque C native qui les rendent bien plus rapides.

8.15.2 Pré-requis

Aucun pré-requis pour utiliser cette extension, hormis la bibliothèque C standard, qui est toujours disponible.

8.15.3 Installation

Depuis PHP 4.2.0, ces fonctions sont activées par défaut. Pour les versions plus anciennes, vous devrez configurer et compiler PHP avec l'option --enable-ctype . Vous pouvez toujours désactiver le support de l'extension ctype avec l'option de compilation --disable-ctype .

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de bibliothèque supplémentaire pour disposer de ces fonctions.

Note

Cette bibliothèque fait partie intégrante de PHP depuis la version 4.3.0.

8.15.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.15.5 Types de ressources

Cette extension ne définit aucune ressource.

8.15.6 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [ctype_alnum](#)
- [ctype_alpha](#)
- [ctype_cntrl](#)
- [ctype_digit](#)
- [ctype_graph](#)
- [ctype_lower](#)
- [ctype_print](#)
- [ctype_punct](#)
- [ctype_space](#)
- [ctype_upper](#)
- [ctype_xdigit](#)

8.15.8 ctype_alpha() : Vérifie qu'une chaîne est alphabétique

bool **ctype_alpha** (string text)

[ctype_alpha](#) vérifie si tous les caractères de la chaîne text sont des lettres, et FALSE sinon. En langage C , les lettres sont [A-Za-z] et [ctype_alpha](#) est équivalent à (ctype_upper(\$text) || ctype_lower(\$text)) , si \$text est un caractère unique, mais certaines langues ont des caractères qui ne sont ni des majuscules, ni des minuscules.

8.15.9 ctype_cntrl() : Vérifie qu'un caractère est un caractère de contrôle

bool **ctype_cntrl** (string text)

[ctype_cntrl](#) vérifie si tous les caractères de la chaîne text sont des caractères de contrôle spéciaux. Ces caractères sont les nouvelles lignes, les tabulations, etc.

8.15.10 ctype_digit() : Vérifie qu'une chaîne est un entier

bool **ctype_digit** (string text)

[ctype_digit](#) vérifie si tous les caractères de la chaîne text sont des chiffres.

8.15.11 `ctype_graph()` : Vérifie qu'une chaîne est imprimable

`bool ctype_graph (string text)`

`ctype_graph` vérifie si tous les caractères de la chaîne text ont une représentation graphique, et créeront une impression sur l'écran (les espaces n'en font pas partie).

8.15.12 `ctype_lower()` : Vérifie qu'une chaîne est en minuscules

`bool ctype_lower (string text)`

`ctype_lower` vérifie si tous les caractères de la chaîne text sont des minuscules.

8.15.13 `ctype_print()` : Vérifie qu'une chaîne est imprimable

`bool ctype_print (string text)`

`ctype_print` vérifie si tous les caractères de la chaîne text vont être imprimables à l'écran (y compris les espaces blancs).

8.15.14 `ctype_punct()` : Vérifie qu'une chaîne contient de la ponctuation

`bool ctype_punct (string text)`

`ctype_punct` vérifie si tous les caractères de la chaîne text sont imprimables, mais ne sont ni des lettres, ni des chiffres, ni des espaces.

8.15.15 `ctype_space()` : Vérifie qu'une chaîne n'est faite que de caractères blancs

`bool ctype_space (string text)`

`ctype_space` vérifie si tous les caractères de la chaîne text vont créer un espace blanc.

8.15.16 `ctype_upper()` : Vérifie qu'une chaîne est en majuscules

`bool ctype_upper (string text)`

`ctype_upper` vérifie si tous les caractères de la chaîne text sont des majuscules.

8.15.17 `ctype_xdigit()` : Vérifie qu'un caractère représente un nombre hexadécimal

`bool ctype_xdigit (string text)`

`ctype_xdigit` vérifie si tous les caractères de la chaîne text sont des chiffres hexadécimaux.

8.16 CURL

8.16.1 Introduction

PHP supporte libcurl, une bibliothèque créée par Daniel Stenberg, qui vous permet de vous connecter, de communiquer avec de nombreux serveurs, grâce à de nombreux protocoles. libcurl supporte actuellement les protocoles suivants : HTTP , HTTPS , FTP , gopher , telnet , dict , file , et LDAP . libcurl supporte aussi les certificats HTTPS, les POST HTTP, PUT HTTP, le chargement par FTP (ce qui peut être fait par l'extension FTP), les chargements par formulaire HTTP, les proxies, les cookies et l'identification par mot de passe et nom d'utilisateur.

Ces fonctions ont été ajoutées en PHP 4.0.2.

8.16.2 Pré-requis

Pour pouvoir utiliser les fonctions de PHP CURL, vous devez installer le paquetage [libcurl](#) . PHP requiert la version libcurl 7.0.2-beta ou plus récente. En PHP 4.2.3, vous aurez besoin de libcurl version 7.9.0 ou plus. A partir de PHP 4.3.0, vous aurez besoin de libcurl version 7.9.8 ou plus. PHP 5.0.0 requiert une version de libcurl supérieure ou égale à 7.10.5.

8.16.3 Installation

Pour utiliser CURL depuis les scripts PHP, vous devez compiler celui-ci avec l'option `--with-curl[=DIR]` où DIR est le chemin jusqu'au dossier contenant les dossiers lib et include . Dans le dossier include il doit se trouver un dossier appelé curl , qui contient notamment les fichiers easy.h et curl.h . Il doit aussi se trouver un fichier nommé libcurl.a dans le dossier lib . A partir de PHP 4.3.0, vous pouvez aussi configurer PHP pour qu'il utilise curl comme gestionnaire d'URL avec l'option `--with-curlwrappers` .

Note

Note aux utilisateurs Win32

Afin d'activer ce module dans l'environnement Windows, vous devez copier les bibliothèques libeay32.dll et sslsleay32.dll depuis le dossier DLL de PHP/Win32 dans le dossier SYSTEM de votre machine. (par exemple : C:\WINNT\SYSTEM32 ou C:\WINDOWS\SYSTEM).

8.16.4 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

- CURLOPT_AUTOREFERER ([entier](#))
Disponible depuis PHP 5.1.0
- CURLOPT_COOKIESESSION ([entier](#))
Disponible depuis PHP 5.1.0
- CURLOPT_DNS_USE_GLOBAL_CACHE ([entier](#))
- CURLOPT_DNS_CACHE_TIMEOUT ([entier](#))

CURLOPT_FTPSSLAUTH (entier)

Disponible depuis PHP 5.1.0

CURLOPT_PORT (entier)

CURLOPT_FILE (entier)

CURLOPT_INFILE (entier)

CURLOPT_INFILESIZE (entier)

CURLOPT_URL (entier)

CURLOPT_PROXY (entier)

CURLOPT_VERBOSE (entier)

CURLOPT_HEADER (entier)

CURLOPT_HTTPHEADER (entier)

CURLOPT_NOPROGRESS (entier)

CURLOPT_NOBODY (entier)

CURLOPT_FAILONERROR (entier)

CURLOPT_UPLOAD (entier)

CURLOPT_POST (entier)

CURLOPT_FTPLISTONLY (entier)

CURLOPT_FTPAPPEND (entier)

CURLOPT_NETRC (entier)

CURLOPT_FOLLOWLOCATION (entier)

CURLOPT_FTPASCII (entier)

CURLOPT_PUT (entier)

CURLOPT_MUTE (entier)

CURLOPT_USERPWD (entier)

CURLOPT_PROXYUSERPWD (entier)

CURLOPT_RANGE (entier)

CURLOPT_TIMEOUT (entier)

CURLOPT_POSTFIELDS (entier)

CURLOPT_REFERER (entier)
CURLOPT_USERAGENT (entier)
CURLOPT_FTPPORT (entier)
CURLOPT_FTP_USE_EPSV (entier)
CURLOPT_LOW_SPEED_LIMIT (entier)
CURLOPT_LOW_SPEED_TIME (entier)
CURLOPT_RESUME_FROM (entier)
CURLOPT_COOKIE (entier)
CURLOPT_SSLCERT (entier)
CURLOPT_SSLCERTPASSWD (entier)
CURLOPT_WRITEHEADER (entier)
CURLOPT_SSL_VERIFYHOST (entier)
CURLOPT_COOKIEFILE (entier)
CURLOPT_SSLVERSION (entier)
CURLOPT_TIMECONDITION (entier)
CURLOPT_TIMEVALUE (entier)
CURLOPT_CUSTOMREQUEST (entier)
CURLOPT_STDERR (entier)
CURLOPT_TRANSFERTEXT (entier)
CURLOPT_RETURNTRANSFER (entier)
CURLOPT_QUOTE (entier)
CURLOPT_POSTQUOTE (entier)
CURLOPT_INTERFACE (entier)
CURLOPT_KRB4LEVEL (entier)
CURLOPT_HTTPPROXYTUNNEL (entier)
CURLOPT_FILETIME (entier)
CURLOPT_WRITEFUNCTION (entier)

CURLOPT_READFUNCTION ([entier](#))
CURLOPT_PASSWDFUNCTION ([entier](#))
CURLOPT_HEADERFUNCTION ([entier](#))
CURLOPT_MAXREDIRS ([entier](#))
CURLOPT_MAXCONNECTS ([entier](#))
CURLOPT_CLOSEPOLICY ([entier](#))
CURLOPT_FRESH_CONNECT ([entier](#))
CURLOPT_FORBID_REUSE ([entier](#))
CURLOPT_RANDOM_FILE ([entier](#))
CURLOPT_EGDSOCKET ([entier](#))
CURLOPT_CONNECTTIMEOUT ([entier](#))
CURLOPT_SSL_VERIFYPEER ([entier](#))
CURLOPT_CAINFO ([entier](#))
CURLOPT_CAPATH ([entier](#))
CURLOPT_COOKIEJAR ([entier](#))
CURLOPT_SSL_CIPHER_LIST ([entier](#))
CURLOPT_BINARYTRANSFER ([entier](#))
CURLOPT_NOSIGNAL ([entier](#))
CURLOPT_PROXYTYPE ([entier](#))
CURLOPT_BUFFERSIZE ([entier](#))
CURLOPT_HTTPGET ([entier](#))
CURLOPT_HTTP_VERSION ([entier](#))
CURLOPT_SSLKEY ([entier](#))
CURLOPT_SSLKEYTYPE ([entier](#))
CURLOPT_SSLKEYPASSWD ([entier](#))
CURLOPT_SSLENGINE ([entier](#))
CURLOPT_SSLENGINE_DEFAULT ([entier](#))

CURLOPT_SSLCERTTYPE (entier)
CURLOPT_CRLF (entier)
CURLOPT_ENCODING (entier)
CURLOPT_PROXYPORT (entier)
CURLOPT_UNRESTRICTED_AUTH (entier)
CURLOPT_FTP_USE_EPRT (entier)
CURLOPT_HTTP200ALIASES (entier)
CURLOPT_HTTPAUTH (entier)
CURLAUTH_BASIC (entier)
CURLAUTH_DIGEST (entier)
CURLAUTH_GSSNEGOTIATE (entier)
CURLAUTH_NTLM (entier)
CURLAUTH_ANY (entier)
CURLAUTH_ANYSAFE (entier)
CURLOPT_PROXYAUTH (entier)
CURLCLOSEPOLICY_LEAST_RECENTLY_USED (entier)
CURLCLOSEPOLICY_LEAST_TRAFFIC (entier)
CURLCLOSEPOLICY_SLOWEST (entier)
CURLCLOSEPOLICY_CALLBACK (entier)
CURLCLOSEPOLICY_OLDEST (entier)
CURLINFO_EFFECTIVE_URL (entier)
CURLINFO_HTTP_CODE (entier)
CURLINFO_HEADER_SIZE (entier)
CURLINFO_REQUEST_SIZE (entier)
CURLINFO_TOTAL_TIME (entier)
CURLINFO_NAMELOOKUP_TIME (entier)
CURLINFO_CONNECT_TIME (entier)

CURLINFO_PRETRANSFER_TIME (entier)
CURLINFO_SIZE_UPLOAD (entier)
CURLINFO_SIZE_DOWNLOAD (entier)
CURLINFO_SPEED_DOWNLOAD (entier)
CURLINFO_SPEED_UPLOAD (entier)
CURLINFO_FILETIME (entier)
CURLINFO_SSL_VERIFYRESULT (entier)
CURLINFO_CONTENT_LENGTH_DOWNLOAD (entier)
CURLINFO_CONTENT_LENGTH_UPLOAD (entier)
CURLINFO_STARTTRANSFER_TIME (entier)
CURLINFO_CONTENT_TYPE (entier)
CURLINFO_REDIRECT_TIME (entier)
CURLINFO_REDIRECT_COUNT (entier)
CURL_VERSION_IPV6 (entier)
CURL_VERSION_KERBEROS4 (entier)
CURL_VERSION_SSL (entier)
CURL_VERSION_LIBZ (entier)
CURLVERSION_NOW (entier)
CURLE_OK (entier)
CURLE_UNSUPPORTED_PROTOCOL (entier)
CURLE_FAILED_INIT (entier)
CURLE_URL_MALFORMAT (entier)
CURLE_URL_MALFORMAT_USER (entier)
CURLE_COULDNT_RESOLVE_PROXY (entier)
CURLE_COULDNT_RESOLVE_HOST (entier)
CURLE_COULDNT_CONNECT (entier)
CURLE_FTP_WEIRD_SERVER_REPLY (entier)

CURLE_FTP_ACCESS_DENIED (entier)
CURLE_FTP_USER_PASSWORD_INCORRECT (entier)
CURLE_FTP_WEIRD_PASS_REPLY (entier)
CURLE_FTP_WEIRD_USER_REPLY (entier)
CURLE_FTP_WEIRD_PASV_REPLY (entier)
CURLE_FTP_WEIRD_227_FORMAT (entier)
CURLE_FTP_CANT_GET_HOST (entier)
CURLE_FTP_CANT_RECONNECT (entier)
CURLE_FTP_COULDNT_SET_BINARY (entier)
CURLE_PARTIAL_FILE (entier)
CURLE_FTP_COULDNT_RETR_FILE (entier)
CURLE_FTP_WRITE_ERROR (entier)
CURLE_FTP_QUOTE_ERROR (entier)
CURLE_HTTP_NOT_FOUND (entier)
CURLE_WRITE_ERROR (entier)
CURLE_MALFORMAT_USER (entier)
CURLE_FTP_COULDNT_STOR_FILE (entier)
CURLE_READ_ERROR (entier)
CURLE_OUT_OF_MEMORY (entier)
CURLE_OPERATION_TIMEOUTED (entier)
CURLE_FTP_COULDNT_SET_ASCII (entier)
CURLE_FTP_PORT_FAILED (entier)
CURLE_FTP_COULDNT_USE_REST (entier)
CURLE_FTP_COULDNT_GET_SIZE (entier)
CURLE_HTTP_RANGE_ERROR (entier)
CURLE_HTTP_POST_ERROR (entier)
CURLE_SSL_CONNECT_ERROR (entier)

CURLE_FTP_BAD_DOWNLOAD_RESUME ([entier](#))
CURLE_FILE_COULDNT_READ_FILE ([entier](#))
CURLE_LDAP_CANNOT_BIND ([entier](#))
CURLE_LDAP_SEARCH_FAILED ([entier](#))
CURLE_LIBRARY_NOT_FOUND ([entier](#))
CURLE_FUNCTION_NOT_FOUND ([entier](#))
CURLE_ABORTED_BY_CALLBACK ([entier](#))
CURLE_BAD_FUNCTION_ARGUMENT ([entier](#))
CURLE_BAD_CALLING_ORDER ([entier](#))
CURLE_HTTP_PORT_FAILED ([entier](#))
CURLE_BAD_PASSWORD_ENTERED ([entier](#))
CURLE_TOO_MANY_REDIRECTS ([entier](#))
CURLE_UNKNOWN_TELNET_OPTION ([entier](#))
CURLE_TELNET_OPTION_SYNTAX ([entier](#))
CURLE_OBSOLETE ([entier](#))
CURLE_SSL_PEER_CERTIFICATE ([entier](#))
CURLE_GOT_NOTHING ([entier](#))
CURLE_SSL_ENGINE_NOTFOUND ([entier](#))
CURLE_SSL_ENGINE_SETFAILED ([entier](#))
CURLE_SEND_ERROR ([entier](#))
CURLE_RECV_ERROR ([entier](#))
CURLE_SHARE_IN_USE ([entier](#))
CURLE_SSL_CERTPROBLEM ([entier](#))
CURLE_SSL_CIPHER ([entier](#))
CURLE_SSL_CACERT ([entier](#))
CURLE_BAD_CONTENT_ENCODING ([entier](#))
CURLE_LDAP_INVALID_URL ([entier](#))

CURLE_FILESIZE_EXCEEDED ([entier](#))

CURLE_FTP_SSL_FAILED ([entier](#))

CURLFTPAUTH_DEFAULT ([entier](#))
Disponible depuis PHP 5.1.0

CURLFTPAUTH_SSL ([entier](#))
Disponible depuis PHP 5.1.0

CURLFTPAUTH_TLS ([entier](#))
Disponible depuis PHP 5.1.0

CURLPROXY_HTTP ([entier](#))

CURLPROXY_SOCKS5 ([entier](#))

CURL_NETRC_OPTIONAL ([entier](#))

CURL_NETRC_IGNORED ([entier](#))

CURL_NETRC_REQUIRED ([entier](#))

CURL_HTTP_VERSION_NONE ([entier](#))

CURL_HTTP_VERSION_1_0 ([entier](#))

CURL_HTTP_VERSION_1_1 ([entier](#))

CURLM_CALL_MULTI_PERFORM ([entier](#))

CURLM_OK ([entier](#))

CURLM_BAD_HANDLE ([entier](#))

CURLM_BAD_EASY_HANDLE ([entier](#))

CURLM_OUT_OF_MEMORY ([entier](#))

CURLM_INTERNAL_ERROR ([entier](#))

CURLMSG_DONE ([entier](#))

8.16.5 Exemples

Une fois que vous avez compilé PHP avec le support CURL, vous pouvez commencer à l'exploiter avec vos scripts PHP. Le principe de fonctionnement est d'initialiser une session CURL avec [curl_init](#) , puis de choisir toutes vos options de transfert avec [curl_setopt](#) , puis d'exécuter la session avec [curl_exec](#) et de finir votre session avec [curl_close](#) . Voici un exemple d'utilisation des fonctions CURL, qui récupère la page principale de [example.com](#) dans un fichier :

Utilisation de CURL pour récupérer une page

<?php

```

$ch = curl_init("http://www.example.com/");
$fp = fopen("example_homepage.txt", "w");

curl_setopt($ch, CURLOPT_FILE, $fp);
curl_setopt($ch, CURLOPT_HEADER, 0);

curl_exec($ch);
curl_close($ch);
fclose($fp);
?>

```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [curl_close](#)
- [curl_copy_handle](#)
- [curl_errno](#)
- [curl_error](#)
- [curl_exec](#)
- [curl_getinfo](#)
- [curl_init](#)
- [curl_multi_add_handle](#)
- [curl_multi_close](#)
- [curl_multi_exec](#)
- [curl_multi_getcontent](#)
- [curl_multi_info_read](#)
- [curl_multi_init](#)
- [curl_multi_remove_handle](#)
- [curl_multi_select](#)
- [curl_setopt](#)
- [curl_version](#)

8.16.7 curl_copy_handle() : Copie une ressource cURL avec toutes ses préférences

resource **curl_copy_handle** (resource ch)

Cette fonction copie une ressource cURL, retournant une nouvelle ressource cURL avec les mêmes préférences.

Copie d'une ressource cURL

```

<?php
// crée une nouvelle ressource curl
$ch = curl_init();

// assigne URL et autres options appropriées
curl_setopt($ch, CURLOPT_URL, 'http://www.example.com/');
curl_setopt($ch, CURLOPT_HEADER, 0);

// copie la ressource
$ch2 = curl_copy_handle($ch);

// attrape l'URL (http://www.example.com/) et le passe au navigateur

```

```

curl_exec($ch);

// ferme les ressources curl et libère les ressources systèmes
curl_close($ch2);
curl_close($ch);
?>

```

8.16.8 curl_errno() : Retourne un entier représentant le dernier message d'erreur CURL

int **curl_errno** (resource ch)

curl_errno retourne le numéro d'erreur de la dernière opération cURL, avec la ressource ch , ou 0 (zéro) si aucune erreur n'est survenue.

Voir aussi curl_error et les codes erreur cURL .

8.16.9 curl_error() : Retourne une chaîne contenant le dernier message d'erreur CURL

string **curl_error** (resource ch)

curl_error retourne un message clair représentant la dernière erreur qui est survenue avec la ressource ch ou " (la chaîne vide) si aucune erreur n'est survenue.

Voir aussi curl_errno et les codes erreur cURL ..

8.16.10 curl_exec() : Exécute une session CURL

mixed **curl_exec** (resource ch)

curl_exec doit être appelée après l'initialisation et le paramétrage d'une session cURL. Son but est simplement d'exécuter la session cURL définie par ch .

Initialise une session CURL et récupère une page web

```

<?php
// création d'une nouvelle ressource CURL
$ch = curl_init();

// configuration de l'URL et d'autres options
curl_setopt($ch, CURLOPT_URL, "http://www.example.com/");
curl_setopt($ch, CURLOPT_HEADER, 0);

// récupération de l'URL et affichage sur le navigateur
curl_exec($ch);

// fermeture de la session curl
curl_close($ch);
?>

```

Note

Si vous voulez récupérer le résultat dans une variable plutôt que de l'afficher directement, utilisez l'option CURLOPT_RETURNTRANSFER de curl_setopt .

8.16.11 curl_getinfo() : Lit les informations détaillant un transfert CURL

mixed **curl_getinfo** (resource *ch* , int *opt*)

curl_getinfo lit les informations détaillant le transfert identifié par la ressource *ch* . Le paramètre *opt* peut prendre l'une des valeurs suivantes :

- CURLINFO_EFFECTIVE_URL - Dernière URL réelle
- CURLINFO_HTTP_CODE - Dernier code HTTP reçu
- CURLINFO_FILETIME - Date distante du document, et -1 si la date du document distant est inconnue.
- CURLINFO_TOTAL_TIME - Durée de la transaction en secondes pour le dernier transfert
- CURLINFO_NAMELOOKUP_TIME - Durée de résolution du nom de domaine en secondes
- CURLINFO_CONNECT_TIME - Durée d'établissement de la connexion en secondes
- CURLINFO_PRETRANSFER_TIME - Durée en secondes, entre le début de la transaction et de début du transfert de fichiers
- CURLINFO_STARTTRANSFER_TIME - Durée en secondes jusqu'à ce que le premier octet soit sur le point d'être transféré
- CURLINFO_REDIRECT_TIME - Durée en secondes de toutes les étapes de redirection avant que la transaction finale ne soit débutée
- CURLINFO_SIZE_UPLOAD - Nombre total d'octets envoyés
- CURLINFO_SIZE_DOWNLOAD - Nombre total d'octets téléchargés
- CURLINFO_SPEED_DOWNLOAD - Vitesse moyenne de téléchargement
- CURLINFO_SPEED_UPLOAD - Vitesse moyenne d'envoi
- CURLINFO_HEADER_SIZE - Taille des en-têtes reçus
- CURLINFO_REQUEST_SIZE - Taille totale des requêtes envoyées. Actuellement, uniquement pour les requêtes HTTP
- CURLINFO_SSL_VERIFYRESULT - Résultat de la vérification de la certification SSL demandée par CURLOPT_SSL_VERIFYPEER
- CURLINFO_CONTENT_LENGTH_DOWNLOAD - Taille du corps du téléchargement, lu dans l'en-tête Content-Length:
- CURLINFO_CONTENT_LENGTH_UPLOAD - Taille spécifiée de l'envoi.
- CURLINFO_CONTENT_TYPE - Content-Type: du fichier téléchargé. NULL indique que le serveur n'a pas envoyé d'en-tête Content-Type:

Lorsqu'elle est appelée sans le paramètre *opt* , un tableau associatif est retourné avec les éléments suivants :

- url
- content_type
- http_code
- header_size
- request_size
- filetime
- ssl_verify_result
- redirect_count
- total_time
- namelookup_time
- connect_time
- pretransfer_time
- size_upload
- size_download
- speed_download
- speed_upload

- `download_content_length`
- `upload_content_length`
- `starttransfer_time`
- `redirect_time`

8.16.12 `curl_init()` : Initialise une session CURL

resource `curl_init` (*string* `url`)

`curl_init` initialise une nouvelle session et retourne un identifiant de session CURL à utiliser avec les fonctions `curl_setopt` , `curl_exec` et `curl_close` . Si le paramètre optionnel `url` est fourni, alors `CURLOPT_URL` prendra cette valeur. Vous pouvez manuellement fixer cette valeur avec la fonction `curl_setopt` .

Initialiser une session CURL et récupérer une page web

```
<?php
// initialisation de la session
$ch = curl_init();

// configuration des options
curl_setopt($ch, CURLOPT_URL, "http://www.example.com/");
curl_setopt($ch, CURLOPT_HEADER, 0);

// exécution de la session
curl_exec($ch);

// fermeture des ressources
curl_close($ch);
?>
```

Voir aussi `curl_close` et `curl_setopt` .

8.16.13 `curl_multi_add_handle()` : Ajoute une ressource cURL à un cURL multiple

int `curl_multi_add_handle` (resource `mh` , resource `ch`)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Voir aussi `curl_multi_init` , `curl_init` et `curl_multi_remove_handle` .

8.16.14 `curl_multi_close()` : Termine un jeu de sessions cURL

void `curl_multi_close` (resource `mh`)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Voir aussi `curl_multi_init` et `curl_close` .

8.16.15 `curl_multi_exec()` : Exécute les sous-requêtes de la session cURL

int `curl_multi_exec` (resource mh , int still_running)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Voir aussi [curl_multi_init](#) et [curl_exec](#) .

8.16.16 `curl_multi_getcontent()` : Retourne le contenu obtenu avec l'option `CURLOPT_RETURNTRANSFER`

string `curl_multi_getcontent` (resource ch)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Voir aussi [curl_multi_init](#) .

8.16.17 `curl_multi_info_read()` : Lit les informations sur les transferts actuels

array `curl_multi_info_read` (resource mh)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Voir aussi [curl_multi_init](#) .

8.16.18 `curl_multi_init()` : Retourne un nouveau cURL multiple

resource `curl_multi_init` ()

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Voir aussi [curl_init](#) et [curl_multi_close](#) .

8.16.19 `curl_multi_remove_handle()` : Retire un cURL multiple d'un jeu de cURL

int `curl_multi_remove_handle` (resource mh , resource ch)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Voir aussi [curl_multi_init](#) , [curl_init](#) et [curl_multi_add_handle](#) .

8.16.20 curl_multi_select() : Retourne les sockets associées à cURL, pour utilisation

int **curl_multi_select** (resource mh , float timeout)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.
Voir aussi curl_multi_init .

8.16.21 curl_setopt() : Modifie une option de transfert CURL

bool **curl_setopt** (resource ch , int option , mixed value)

`curl_setopt` fixe les options de transfert de la session CURL identifiée par `ch` . `option` est le nom de l'option à fixer, et `value` est sa valeur.

`value` doit être un booléen pour les valeurs suivantes du paramètres `option` :

Option	Définissez le paramètre <code>value</code> à	Notes
CURLOPT_AUTOREFERER	TRUE pour fixer automatiquement le champ Referer: dans les requêtes où une redirection Location: suit.	Disponible depuis PHP 5.1.0.
CURLOPT_BINARYTRANSFER	TRUE pour retourner la ligne affichée lorsque CURLOPT_RETURNTRANSFER est utilisé.	
CURLOPT_COOKIESESSION	TRUE pour marquer ceci comme un nouveau cookie "session". Cela forcera libcurl à ignorer tous les cookies qui sont prêts à être chargés qui sont des "cookies de session" provenant des sessions antérieures. Par défaut, libcurl enregistre et charge toujours tous les cookies, indépendamment s'ils sont des cookies de session ou pas. Les cookies de session sont des cookies sans date d'expiration et existeront que pour cette "session" seulement.	Disponible depuis PHP 5.1.0.
CURLOPT_CRLF	TRUE pour convertir les nouvelles lignes Unix en nouvelles lignes CRLF pendant le transfert.	
CURLOPT_DNS_USE_GLOBAL_CACHE	TRUE pour utiliser un cache DNS global. Cette option n'est pas thread-safe et est activée par défaut.	
CURLOPT_FAILONERROR	TRUE pour que PHP traite silencieusement les codes HTTP supérieurs à 300. Le comportement par défaut est de retourner la page normalement, en ignorant ce code.	
CURLOPT_FILETIME	TRUE pour tenter de récupérer la date de modification du document distant. Vous pouvez également retrouver cette valeur en utilisant l'option CURLINFO_FILETIME	

	avec <code>curl_getinfo</code> .	
CURLOPT_FOLLOWLOCATION	TRUE pour suivre toutes les en-têtes "Location: " que le serveur envoie dans les en-têtes HTTP (notez que cette fonction est récursive et que PHP suivra toutes les en-têtes "Location: " qu'il trouvera à moins que CURLOPT_MAXREDIRS ne soit définie).	
CURLOPT_FORBID_REUSE	TRUE pour forcer la connexion à se fermer explicitement lorsque le processus est terminé et ne sera pas mise en cache pour être réutilisée.	
CURLOPT_FRESH_CONNECT	TRUE pour forcer à utiliser une nouvelle connexion au lieu de celle en cache.	
CURLOPT_FTP_USE_EPRT	TRUE pour utiliser EPRT (et LPRT) lors de l'activation des téléchargements FTP. Utilisez FALSE pour désactiver EPRT et LPRT et ainsi, n'utiliser que PORT.	Ajouté en PHP 5.0.0.
CURLOPT_FTP_USE_EPSV	TRUE pour tenter tout d'abord une commande EPSV pour les transferts FTP avant de revenir à une commande PASV. Définissez cette option à FALSE pour désactiver EPSV.	
CURLOPT_FTPAPPEND	TRUE pour que PHP concatène le fichier distant, plutôt que de l'écraser.	
CURLOPT_FTPASCII	Un alias de CURLOPT_TRANSFERTEXT . Utilisez celui-là à la place.	
CURLOPT_FTPLISTONLY	TRUE pour ne lister que les noms d'un dossier FTP. directory.	
CURLOPT_HEADER	TRUE pour inclure l'en-tête dans la valeur de retour.	
CURLOPT_HTTPGET	TRUE pour réinitialiser la méthode de requête HTTP à POST. Sachant que GET est la valeur par défaut, cela n'est uniquement nécessaire que si la méthode de requête a été changée.	
CURLOPT_HTTPPROXYTUNNEL	TRUE pour effectuer un tunnel à travers un proxy HTTP.	
CURLOPT_MUTE	TRUE pour que PHP soit totalement silencieux concernant toutes les fonctions CURL.	
CURLOPT_NETRC	TRUE pour que PHP analyse votre fichier ~/.netrc et utilise votre nom de compte et mot de passe sur le site distant que vous souhaitez contacter.	
CURLOPT_NOBODY	TRUE pour que le corps du transfert ne soit pas inclus dans la valeur de retour.	
CURLOPT_NOPROGRESS	TRUE pour désactiver la barre de progression des transferts CURL. transferts.	
	Note	

	PHP choisit automatiquement une valeur non nulle. Ne changez cette valeur que le temps du débogage.	
CURLOPT_NOSIGNAL	TRUE pour ignorer toutes les fonctions CURL qui cause l'envoi d'un signal au processus PHP. Activé par défaut dans les SAPIs multi-threadés, les options de timeout peut toujours être utilisées.	Ajouté dans CURL 7.10 et PHP 5.0.0.
CURLOPT_POST	TRUE pour que PHP fasse un HTTP POST. Un POST est un encodage normal "application/x-www-form-urlencoded", utilisé couramment par les formulaires HTML.	
CURLOPT_PUT	TRUE pour que le chargement se fasse par HTTP PUT. Le fichier à charger doit être fixé avec les options CURLOPT_INFILE et CURLOPT_INFILESIZE .	
CURLOPT_RETURNTRANSFER	TRUE retourne directement le transfert sous forme de chaîne de la valeur retournée par <code>curl_exec</code> au lieu de l'afficher directement.	
CURLOPT_SSL_VERIFYPEER	FALSE pour arrêter CURL de vérifier le certificat. Les certificats alternatifs peuvent être spécifiés avec l'option CURLOPT_CAINFO (ajouté dans CURL 7.9.8) ou un répertoire de certificat peut être spécifié avec l'option CURLOPT_CAPATH . Tout comme CURL 7.10, CURL installe un paquet par défaut. CURLOPT_SSL_VERIFYHOST doit aussi être positionnée à 1 ou 0 si CURLOPT_SSL_VERIFYPEER est désactivée (par défaut à 2).	TRUE par défaut depuis CURL 7.10. Paquet installé par défaut depuis CURL 7.10.
CURLOPT_TRANSFERTEXT	TRUE pour utiliser le mode ASCII pour les transferts FTP. Pour LDAP, il récupère les données en texte plein au lieu d'Html. Sur les systèmes Windows, STDOUT ne sera pas définie en mode binaire.	
CURLOPT_UNRESTRICTED_AUTH	TRUE pour garder l'envoi du nom de l'utilisateur ainsi que le mot de passe lorsque l'on suit les chemins (en utilisant CURLOPT_FOLLOWLOCATION), même si le nom d'hôte change.	Ajouté en PHP 5.0.0.
CURLOPT_UPLOAD	TRUE pour que PHP prépare un chargement.	
CURLOPT_VERBOSE	TRUE pour afficher tous les événements. Ecrit la sortie sur stderr ou dans le fichier spécifié en utilisant CURLOPT_STDERR .	

value doit être un entier pour les valeurs suivantes du paramètres option :

Option	Définissez le paramètre value à	Notes
CURLOPT_BUFFERSIZE		

	La taille du buffer à utiliser pour chaque lecture. Cependant, il n'y a aucune garantie que cette requête soit accomplie.	Ajouté en CURL 7.10 et PHP 5.0.0.
CURLOPT_CLOSEPOLICY	Soit CURLCLOSEPOLICY_LEAST_RECENTLY_USED , soit CURLCLOSEPOLICY_OLDEST . Il y a trois autres constantes CURLCLOSEPOLICY_ mais CURL ne les supporte pas encore.	
CURLOPT_CONNECTTIMEOUT	Le nombre de secondes à attendre durant la tentative de connexion. Utilisez une valeur de 0 pour attendre indéfiniment.	
CURLOPT_DNS_CACHE_TIMEOUT	Le temps en seconde que CURL doit conserver les entrées DNS en mémoire. Cette option est définie à 120 (2minutes) par défaut.	
CURLOPT_FTPSSLAUTH	La méthode d'identification FTP, lorsqu'elle est activée : CURLFTPAUTH_SSL (tente SSL en premier), CURLFTPAUTH_TLS (tente TLS en premier) ou CURLFTPAUTH_DEFAULT (laisse CURL décider).	Ajouté en CURL 7.12.2 et PHP 5.1.0.
CURLOPT_HTTP_VERSION	CURL_HTTP_VERSION_NONE (défaut, laisse CURL décider la version à utiliser), CURL_HTTP_VERSION_1_0 (force HTTP/1.0), ou CURL_HTTP_VERSION_1_1 (force HTTP/1.1).	
CURLOPT_HTTPAUTH	<p>La méthode d'identification HTTP à utiliser. Ces options sont : CURLAUTH_BASIC , CURLAUTH_DIGEST , CURLAUTH_GSSNEGOTIATE , CURLAUTH_NTLM , CURLAUTH_ANY et CURLAUTH_ANYSAFE .</p> <p>Vous pouvez utiliser le séparateur ou un opérateur pour combiner plus d'une méthode. Si vous faites cela, CURL interrogera le serveur pour voir quelles sont les méthodes supportées et prendra la meilleur.</p> <p>CURLAUTH_ANY est un alias pour CURLAUTH_BASIC CURLAUTH_DIGEST CURLAUTH_GSSNEGOTIATE CURLAUTH_NTLM .</p> <p>CURLAUTH_ANYSAFE est un alias pour CURLAUTH_DIGEST CURLAUTH_GSSNEGOTIATE CURLAUTH_NTLM .</p>	Ajouté en PHP 5.0.0.
CURLOPT_INFILESIZE	La taille attendue, en octets, du fichier lors du téléchargement d'un fichier depuis un site distant.	
CURLOPT_LOW_SPEED_LIMIT	La vitesse de transfert minimale en octets par secondes en dessous de laquelle, et pendant CURLOPT_LOW_SPEED_TIME secondes, PHP considérera qu'elle est trop lente, et annulera le	

	transfert.	
CURLOPT_LOW_SPEED_TIME	Le temps en secondes, qui, si la vitesse de transfert reste en dessous de CURLOPT_LOW_SPEED_LIMIT , PHP considèrera que la connexion est trop lente, et l'annulera.	
CURLOPT_MAXCONNECTS	Le nombre maximal de connexions persistantes autorisées. Lorsque la limite est atteinte, l'option CURLOPT_CLOSEPOLICY est utilisé pour afficher les connexions fermées.	
CURLOPT_MAXREDIRS	Le nombre maximal de redirections HTTP à suivre. Utilisez cette option avec l'option CURLOPT_FOLLOWLOCATION .	
CURLOPT_PORT	Le numéro du port de connexion, à la place de la valeur par défaut utilisée par le protocole.	
CURLOPT_PROXYAUTH	La méthode d'identification HTTP à utiliser pour la connexion à un proxy. Utilisez la même méthode que celle décrite dans CURLOPT_HTTPAUTH . Pour une identification avec un proxy, seuls CURLAUTH_BASIC et CURLAUTH_NTLM sont actuellement supportés.	Ajouté en CURL 7.10.7 et PHP 5.1.0.
CURLOPT_PROXYPORT	Le numéro du port du proxy à utiliser pour la connexion. Ce numéro de port peut également être défini dans l'option CURLOPT_PROXY .	Ajouté en PHP 5.0.0.
CURLOPT_PROXYTYPE	Soit CURLPROXY_HTTP (par défaut), soit CURLPROXY_SOCKS5 .	Ajouté en CURL 7.10 et PHP 5.0.0.
CURLOPT_RESUME_FROM	L'offset, en octets, à partir duquel vous voulez commencer le transfert.	
CURLOPT_SSL_VERIFYHOST	1 pour vérifier l'existence d'un nom commun dans le certificat SSL. 2 pour vérifier l'existence d'un nom commun et vérifier qu'il correspond avec le nom d'hôte fourni.	
CURLOPT_SSLVERSION	La version de SSL (2 ou 3) à utiliser. Par défaut, PHP essaiera de le déterminer lui-même, bien que dans certains cas, il vous faudra le faire manuellement.	
CURLOPT_TIMECONDITION	Comment CURLOPT_TIMEVALUE est traité. Utilisez CURL_TIMECOND_IFMODSINCE pour retourner la page uniquement si elle a été modifiée depuis le temps spécifié par CURLOPT_TIMEVALUE . Si elle n'a pas été modifiée, un en-tête HTTP "304 Not Modified" sera retourné en supposant que CURLOPT_HEADER vaut TRUE . Utilisez CURL_TIMECOND_ISUNMODSINCE pour faire l'inverse. CURL_TIMECOND_IFMODSINCE est par défaut.	Ajouté en PHP 5.1.0.
CURLOPT_TIMEOUT	Le temps maximum d'exécution de la fonction CURL.	

CURLOPT_TIMEVALUE	Le temps en secondes depuis le 1er janvier 1970. Cette valeur sera utilisée comme spécifié dans l'option CURLOPT_TIMECONDITION . Par défaut, CURL_TIMECOND_IFMODSINCE sera utilisée.
-------------------	--

value doit être une chaîne pour les valeurs suivantes du paramètres option :

Option	Définissez le paramètre value à	Notes		
CURLOPT_CAINFO	Le nom d'un fichier contenant un ou plusieurs certificats pour vérifier la concordance. Cela n'a de sens que si vous l'utilisez en combinaison de l'option CURLOPT_SSL_VERIFYPEER .			
CURLOPT_CAPATH	Un dossier qui contient les certificats CA. Utilisez cette option avec l'option CURLOPT_SSL_VERIFYPEER .			
CURLOPT_COOKIE	Le contenu de l'en-tête "Set-Cookie: " , à transmettre dans l'en-tête HTTP.			
CURLOPT_COOKIEFILE	Le nom du fichier contenant les données de cookie. Le fichier de cookie peut être au format Netscape, ou simplement des en-têtes HTTP écrits dans un fichier.			
CURLOPT_COOKIEJAR	Le nom de fichier pour y sauvegarder tous les cookies internes lorsque la connexion se ferme.			
CURLOPT_CUSTOMREQUEST	<p>Une méthode de requête qui sera utilisée à la place de "GET" ou "HEAD" lors des requêtes HTTP. Cette commande est pratique pour effectuer un "DELETE" ou une autre commande HTTP exotique. Les valeurs valides sont "GET" , "POST" , "CONNECT" et plus ; i.e. n'entrez pas une requête HTTP ici. Pour le moment, entrez "GET /index.html HTTP/1.0\r\n\r\n" serait incorrect.</p> <table border="1"> <tr> <td>Note</td> </tr> <tr> <td>N'utilisez pas cette commande sans vous assurer que le serveur l'accepte.</td> </tr> </table>	Note	N'utilisez pas cette commande sans vous assurer que le serveur l'accepte.	
Note				
N'utilisez pas cette commande sans vous assurer que le serveur l'accepte.				
CURLOPT_EGDSOCKET	Comme CURLOPT_RANDOM_FILE excepté que vous passez une chaîne qui contient un nom de fichier vers le socket Entropy Gathering Daemon .			
CURLOPT_ENCODING	Le contenu des en-têtes "Accept-Encoding: " et active le décodage de la réponse. Les encodages supportés sont "identity" , "deflate" et "gzip" . Si une chaîne vide "" est utilisé, un en-tête contenant tous les types d'encodage supportés est envoyé.	Ajouté en CURL 7.10.		
CURLOPT_FTPPORT	La valeur qui sera utilisée pour récupérer l'adresse IP utilisée pour l'instruction FTP "PORT". L'instruction POST indique au serveur distant de se connecter à cette adresse IP. La chaîne peut être une adresse IP, un nom d'hôte, un nom d'interface réseau (sous UNIX), ou juste '-', pour utiliser les IP par défaut du système.			
CURLOPT_INTERFACE	Le nom de l'interface à utiliser. Cela peut être le nom d'une interface, une adresse IP ou encore le nom de l'hôte.			

CURLOPT_KRB4LEVEL	Le degré de sécurité KRB4 (Kerberos 4). Chacune des valeurs (dans l'ordre du plus petit au plus grand) suivantes sont valides : "clear" , "safe" , "confidential" , "private". . Si la chaîne passée ne correspond pas à une de ces valeurs, la valeur "private" sera définie. Positionner cette valeur à NULL revient à désactiver la sécurité KRB4. Actuellement, la sécurité KRB4 fonctionne uniquement avec les transaction FTP.	
CURLOPT_POSTFIELDS	Toutes les données à passer lors d'une opération de HTTP POST.	
CURLOPT_PROXY	Le nom du proxy HTTP au tunnel qui le demande.	
CURLOPT_PROXYUSERPWD	Un nom d'utilisateur et un mot de passe formatés sous la forme "[username]:[password]" à utiliser pour la connexion avec le proxy.	
CURLOPT_RANDOM_FILE	Un nom de fichier à utiliser pour interroger le générateur de nombre aléatoire pour SSL.	
CURLOPT_RANGE	la plage de valeurs à récupérer sous la forme "X-Y" , où les valeurs de X ou Y peuvent être omises. Le transfert HTTP supporte aussi plusieurs intervalles, séparés par des virgules : "X-Y,N-M" .	
CURLOPT_REFERER	Le contenu de l'en-tête "Referer: " à utiliser dans une requête HTTP.	
CURLOPT_SSL_CIPHER_LIST	Une liste de ciphers à utiliser avec SSL. Par exemple, RC4-SHA et TLSv1 sont des listes valides de cipher.	
CURLOPT_SSLCERT	Le nom de fichier du certificat, au format PEM.	
CURLOPT_SSLCERTPASSWD	Le mot de passe nécessaire pour utiliser le certificat CURLOPT_SSLCERT .	
CURLOPT_SSLCERTTYPE	Le format de votre certificat. Les formats supportés sont "PEM" (default), "DER" , et "ENG" .	Ajouté en CURL 7.9.3 et PHP 5.0.0.
CURLOPT_SSLENGINE	L'identifiant pour le moteur de cryptage de votre clé privée spécifié dans CURLOPT_SSLKEY .	
CURLOPT_SSLENGINE_DEFAULT	L'identifiant pour le moteur de cryptage utilisé pour les opérations de cryptage asymétrique.	
CURLOPT_SSLKEY	Un nom de fichier contenant votre clé privée SSL.	
CURLOPT_SSLKEYPASSWD	Le mot de passe secret utilisé par votre clé SSL privée spécifié dans CURLOPT_SSLKEY .	
	Note Du fait que cette option contient un mot de passe sensible, souvenez-vous de conserver le script PHP qui le contient en toute sécurité.	
CURLOPT_SSLKEYTYPE	Le type de votre clé SSL privée spécifié dans CURLOPT_SSLKEY . Les types de clés supportés sont "PEM" (défaut), "DER" , et "ENG" .	
CURLOPT_URL	L'URL à récupérer. Vous pouvez aussi choisir cette	

	valeur lors de l'appel à <code>curl_init</code> .
<code>CURLOPT_USERAGENT</code>	Le contenu de l'en-tête "User-Agent: " à utiliser dans une requête HTTP.
<code>CURLOPT_USERPWD</code>	Un nom d'utilisateur et un mot de passe formatés sous la forme "[username]:[password]" à utiliser pour la connexion.

value doit être un tableau pour les valeurs suivantes du paramètres option :

Option	Définissez le paramètre value à	Notes
<code>CURLOPT_HTTP200ALIASES</code>	Un tableau de réponses HTTP 200 qui sera traité comme réponses valides et non comme des erreurs.	Ajouté en CURL 7.10.3 et PHP 5.0.0.
<code>CURLOPT_HTTPHEADER</code>	Un tableau de champs d'en-têtes HTTP à définir.	
<code>CURLOPT_POSTQUOTE</code>	Un tableau de commandes FTP à exécuter sur le serveur après que la requête FTP se soit exécutée.	
<code>CURLOPT_QUOTE</code>	Un tableau de commande FTP à exécuter sur le serveur avant la requête FTP.	

value doit être une ressource (utilisant `fopen` , par exemple) pour les valeurs suivantes du paramètres option :

Option	Définissez le paramètre value à	Notes
<code>CURLOPT_FILE</code>	Le fichier où sera écrit le transfert. Par défaut, STDOUT (la fenêtre du navigateur).	
<code>CURLOPT_INFILE</code>	Le fichier lu par le transfert lors du chargement.	
<code>CURLOPT_STDERR</code>	Un chemin alternatif à utiliser pour afficher les erreurs au lieu de STDERR .	
<code>CURLOPT_WRITEHEADER</code>	Le fichier où sera écrit les parties d'en-tête du transfert.	

value doit être une chaîne de caractères qui sera un nom valide de fonction de callback pour les valeurs suivantes du paramètres option :

Option	Définissez le paramètre value à	Notes
<code>CURLOPT_HEADERFUNCTION</code>	Le nom d'une fonction de callback où la fonction de callback prend deux paramètres. Le premier est la ressource CURL, le second, une chaîne contenant les données de l'en-tête à écrire. En utilisant cette fonction de callback, il en devient de votre responsabilité de l'écriture des données de l'en-tête. Retourne le nombre d'octets écrits.	
<code>CURLOPT_PASSWDFUNCTION</code>	Le nom d'une fonction de callback où la fonction de callback prend trois paramètres. Le premier est la ressource CURL, le second, une chaîne contenant un mot de passe de prompt et le troisième, est la longueur maximale du mot de passe. Retourne une chaîne contenant le mot de passe.	
<code>CURLOPT_READFUNCTION</code>	Le nom d'une fonction de callback où la fonction de callback prend deux paramètres. Le premier est la ressource CURL et le second, une chaîne contenant les données à lire. En utilisant cette fonction de callback, il devient de votre responsabilité de lire les données.	

	Retourne le nombre d'octets lus. Retourne 0 à un signal EOF .
CURLOPT_WRITEFUNCTION	Le nom d'une fonction de callback où la fonction de callback prend deux paramètres. Le premier est la ressource CURL et le second, une chaîne contenant les données à écrire. En utilisant cette fonction de callback, il devient de votre responsabilité d'écrire les données. Doit retourner le nombre exact d'octets écrits ou il échoua.

Initialisation d'une nouvelle session CURL et recherche d'une page Web

```
<?php
// Création d'une ressource CURL
$ch = curl_init();

// Définition de l'URL et autres options appropriées
curl_setopt($ch, CURLOPT_URL, "http://www.example.com/");
curl_setopt($ch, CURLOPT_HEADER, false);

// Récupération de l'URL et passage au navigateur
curl_exec($ch);

// Fermeture de la ressource CURL et libération des ressources systèmes
curl_close($ch);
?>
```

8.16.22 curl_version() : Retourne la version courante de CURL

array **curl_version** (*int version*)

curl_version retourne une chaîne avec la version courante de la bibliothèque CURL.

8.17 Paiement Cybercash

8.17.1 Installation

Ces fonctions ne sont disponibles que si PHP a été compilé avec l'option `--with-cybercash=[DIR]` .

Cette extension a été déplacée depuis la distribution PHP 4.3.0 dans [PECL](#) .

Si vous avez des questions concernant le statut de CyberCash, voyez la [FAQ CyberCash](#) . En bref, Cybercash a été acheté par VeriSign et bien que le service CyberCash existe toujours, VeriSign encourage les clients à changer. Voyez la FAQ ci-dessus et le lien sur PECL pour plus de détails.

- [Installation](#)
- [cybercash_base64_decode](#)
- [cybercash_base64_encode](#)
- [cybercash_decr](#)
- [cybercash_encr](#)

8.17.3 `cybercash_base64_encode()` : Encodage en base64 pour Cybercash

string `cybercash_base64_encode` (string inbuff)

8.17.4 `cybercash_decr()` : Décryptage Cybercash

array `cybercash_decr` (string wmk , string sk , string inbuff)

`cybercash_decr` retourne un tableau associatif, contenant les éléments "errcode" et, si "errcode" vaut FALSE , "outbuff" (string), "outLth" (long) et "macbuff" (string).

8.17.5 `cybercash_encr()` : Encryptage Cybercash

array `cybercash_encr` (string wmk , string sk , string inbuff)

`cybercash_encr` retourne un tableau associatif, contenant les éléments "errcode" et, si "errcode" vaut FALSE , "outbuff" (string), "outLth" (long) et "macbuff" (string).

8.18 Fonctions Crédit Mutuel CyberMUT

8.18.1 Introduction

Cette extension vous permet de faire des transactions avec des cartes de crédits en utilisant le système Crédit Mutuel CyberMUT .

CyberMUT est un service de paiement web en France, fournie par la banque le Crédit Mutuel. Si vous n'êtes pas en France, ces fonctions ne seront pas utiles pour vous.

L'utilisation de ces fonctions est identiques aux fonctions d'origine, excepté pour les paramètres de retour de `CreerFormulaireCM` et `CreerReponseCM`, qui sont directement retournés par les fonctions PHP, lorsqu'ils sont passés par référence dans les fonctions originales.

Note
Cette extension n'est pas disponible sur les plates-formes Windows.
Note
Cette extension a été déplacée dans le module <u>PECL</u> et ne sera plus intégrée dans PHP à partir de PHP 4.3.0

8.18.2 Pré-requis

Vous devez avoir le SDK approprié pour votre plate-forme, qui doit vous être envoyé après votre inscription CyberMUT (contactez-les par le web ou allez dans un Crédit Mutuel le plus proche).

Note
Ces fonctions fournissent uniquement un lien vers le SDK CyberMUT. Assurez-vous de lire le guide du développeur CyberMUT pour plus de détails sur les paramètres requis.

- [Introduction](#)
- [Pré-requis](#)
- [cybermut_creerformulairecm](#)
- [cybermut_creerreponsecm](#)
- [cybermut_testmac](#)

8.18.4 `cybermut_creerreponsecm()` : Génère le remerciement lors de la confirmation du paiement

string `cybermut_creerreponsecm` (string sentence)

Le paramètre est "OK" si le message de confirmation du paiement a été correctement identifié par la fonction `cybermut_testmac` . Tout autre chaîne est considéré comme un message d'erreur.

8.18.5 `cybermut_testmac()` : Assurez-vous qu'il n'y avait aucune falsification de données dans le message de confirmation reçu

bool `cybermut_testmac` (string code_mac , string version , string tpe , string cdate , string price , string ref_command , string text_free , string code_return)

cybermut_testmac est utilisé pour vous assurer qu'il n'y a aucune falsification de données dans le message de confirmation reçu. Faîte attention aux paramètres `code_return` et `text_free` , qui ne peuvent pas être évalués tel que, en raison du tiret.

8.19 Cyrus IMAP

8.19.1 Introduction

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.
Note
Cette extension n'est pas disponible sur les plates-formes Windows.

8.19.2 Installation

Pour activer le support Cyrus IMAP et en utiliser les fonctions, vous devez compiler PHP avec l'option `--with-cyrus`.

Attention
L'extension IMAP ne peut être utilisée simultanément avec l'extension recode , l'extension YAZ ou l'extension Cyrus . Ceci est dû au fait que ces bibliothèques utilisent un symbole interne commun.

8.19.3 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

[CYRUS_CONN_NONSYNCLITERAL](#) ([entier](#))
[CYRUS_CONN_INITIALRESPONSE](#) ([entier](#))
[CYRUS_CALLBACK_NUMBERED](#) ([entier](#))
[CYRUS_CALLBACK_NOLITERAL](#) ([entier](#))

- [Introduction](#)
- [Installation](#)
- [Constantes pré-définies](#)
- [cyrus_authenticate](#)
- [cyrus_bind](#)
- [cyrus_close](#)
- [cyrus_connect](#)
- [cyrus_query](#)
- [cyrus_unbind](#)

8.19.5 `cyrus_bind()` : Ajoute une fonction de callback sur une connexion Cyrus IMAP

`bool cyrus_bind (resource connection , array callbacks)`

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.19.6 **cyrus_close()** : Ferme la connexion à un serveur Cyrus IMAP

bool **cyrus_close** (resource connection)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.19.7 **cyrus_connect()** : Se connecte à un serveur Cyrus IMAP

resource **cyrus_connect** (*string* host , *string* port , *int* flags)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.19.8 **cyrus_query()** : Envoie une requête à un serveur Cyrus IMAP

array **cyrus_query** (resource connection , string query)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.19.9 **cyrus_unbind()** : Supprime ...

bool **cyrus_unbind** (resource connection , string trigger_name)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.20 Dates et heures

8.20.1 Introduction

Ces fonctions vous permettent de manipuler la date et l'heure sur le serveur qui exécute PHP. Vous pouvez utiliser ces fonctions pour formater la date et l'heure de nombreuses façons.

Note

N'oubliez pas que ces fonctions dépendent de la configuration locale de votre serveur. Faites notamment attention aux changements d'heure d'hiver et d'été (e.g. utilisez `$date = strtotime('+7 days', $date)` et non pas `$date += 7*24*60*60`) ainsi qu'aux années bissextiles lorsque vous utilisez ces fonctions.

8.20.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.20.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.20.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.20.5 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable	Historique
<code>date.default_latitude</code>	"31.7667"	PHP_INI_ALL	Disponible depuis PHP 5.0.0.
<code>date.default_longitude</code>	"35.2333"	PHP_INI_ALL	Disponible depuis PHP 5.0.0.
<code>date.sunrise_zenith</code>	"90.83"	PHP_INI_ALL	Disponible depuis PHP 5.0.0.
<code>date.sunset_zenith</code>	"90.83"	PHP_INI_ALL	Disponible depuis PHP 5.0.0.
<code>date.timezone</code>	""	PHP_INI_ALL	Disponible depuis PHP 5.1.0.

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

```

date.default_latitude float
    La latitude par défaut.
date.default_longitude float
    La longitude par défaut.
date.sunrise_zenith float
    L'heure de lever du soleil par défaut.
date.sunset_zenith float
    L'heure du coucher du soleil par défaut.
```

date.timezone string

Le décalage horaire utilisé par toutes les fonctions date/heure si la variable d'environnement TZ n'est pas définie. L'ordre de priorité est décrit dans la page [date_default_timezone_get](#) .

Note

Les quatres premières options de configuration sont actuellement utilisées uniquement par les fonction [date_sunrise](#) et [date_sunset](#) .

8.20.6 Types de ressources

Cette extension ne définit aucune ressource.

8.20.7 Constantes pré-définies

Les constantes suivantes sont définies depuis PHP 5.1.1 et offrent une représentation standart des dates, qui peut être utilisée avec toutes les fonctions de formattage de date (comme [date](#)).

DATE_ATOM ([chaîne de caractères](#))

Atom (exemple : 2005-08-15T15:52:01+0000)

DATE_COOKIE ([chaîne de caractères](#))

Cookies HTTP (exemple : Mon, 15 Aug 2005 15:52:01 UTC)

DATE_ISO8601 ([chaîne de caractères](#))

ISO-8601 (exemple : 2005-08-15T15:52:01+0000)

DATE_RFC822 ([chaîne de caractères](#))

RFC 822 (exemple : Mon, 15 Aug 2005 15:52:01 UTC)

DATE_RFC850 ([chaîne de caractères](#))

RFC 850 (exemple : Monday, 15-Aug-05 15:52:01 UTC)

DATE_RFC1036 ([chaîne de caractères](#))

RFC 1036 (exemple : Monday, 15-Aug-05 15:52:01 UTC)

DATE_RFC1123 ([chaîne de caractères](#))

RFC 1123 (exemple : Mon, 15 Aug 2005 15:52:01 UTC)

DATE_RFC2822 ([chaîne de caractères](#))

RFC 2822 (Mon, 15 Aug 2005 15:52:01 +0000)

DATE_RSS ([chaîne de caractères](#))

RSS (Mon, 15 Aug 2005 15:52:01 UTC)

DATE_W3C ([chaîne de caractères](#))

World Wide Web Consortium (W3C) (exemple : 2005-08-15T15:52:01+0000)

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [checkdate](#)
- [date_default_timezone_get](#)
- [date_default_timezone_set](#)
- [date_sunrise](#)
- [date_sunset](#)
- [date](#)
- [getdate](#)

- [gettimeofday](#)
- [gmdate](#)
- [gmmktime](#)
- [gmstrftime](#)
- [idate](#)
- [localtime](#)
- [microtime](#)
- [mktime](#)
- [strftime](#)
- [strtotime](#)
- [strtotime](#)
- [time](#)

8.20.9 `date_default_timezone_get()` : Récupère le décalage horaire par défaut utilisé par toutes les fonctions date/heure dans un script

string `date_default_timezone_get` (void)

Cette fonction retourne le décalage horaire par défaut, utilisant l'ordre suivant :

- Le décalage horaire est définie en utilisant la fonction `date_default_timezone_set` (si elle existe)
- La variable d'environnement TZ (si elle n'est pas vide)
- L'option de configuration `date.timezone` (si elle est définie)
- Interrogation magique (si le système le supporte)
- Si aucune des options suivantes ne réussissent, retourne UTC

8.20.10 `date_default_timezone_set()` : Définit le décalage horaire par défaut utilisé par toutes les fonctions date/heure dans un script

bool `date_default_timezone_set` (string `timezone_identifieur`)

`date_default_timezone_set` définit le décalage horaire par défaut utilisé par toutes les fonctions date/heure.

Note

Depuis PHP 5.1.0 (lorsque les fonctions date/heure ont été écrites), chaque appel à une fonction date/heure génère une E_NOTICE si le décalage horaire n'est pas valide et/ou un message E_STRICT si vous utilisez des configurations système ou la variable d'environnement TZ .

8.20.11 `date_sunrise()` : Retourne l'heure de levé du soleil pour un jour et un endroit donnés

mixed `date_sunrise` (int `timestamp` , int `format` , float `latitude` , float `longitude` , float `zenith` , float `gmt_offset`)

`date_sunrise` retourne l'heure de levé du soleil pour un jour (spécifié par le paramètre `timestamp`) et un endroit donnés. Les paramètres `latitude` , `longitude` et `zenith` ont pour valeurs par défaut, respectivement, les valeurs des options de configuration `date.default_latitude` , `date.default_longitude` et `date.sunrise_zenith` .

La valeur par défaut du paramètre latitude est North . Si vous voulez passer une valeur correspondant au Sud, vous devez donc passer une valeur négative. De même pour le paramètre longitude , qui, par défaut, vaut East .

Le paramètre `gmt_offset` est spécifié en heures.

constante	description	exemple
SUNFUNCS_RET_STRING	retourne le résultat sous forme de <u>chaîne de caractères</u>	16:46
SUNFUNCS_RET_DOUBLE	retourne le résultat sous la forme d'un <u>nombre à virgule flottante</u>	16.78243132
SUNFUNCS_RET_TIMESTAMP	retourne le résultat sous la forme d'un <u>entier</u> (timestamp)	1095034606

Exemple avec `date_sunrise`

```
<?php
/* calcul de l'heure de levé du soleil pour la ville de Lisbonne, Portugal
Latitude: 38.4 North
Longitude: 9 West
Zenith ~= 90
offset: +1 GMT
*/

echo 'Heure de levé du soleil pour le ' . date("d/m/Y"). ' : ' . date_sunrise(time(), SUNFUNCS_RET_TIMESTAMP);
?>
```

L'exemple ci-dessus va afficher quelque chose de similaire à :

Heure de levé du soleil pour le 20/12/2004 : 08:54

Voir aussi [date_sunset](#) .

8.20.12 `date_sunset()` : Retourne l'heure de coucher du soleil pour un jour et un endroit donnés

mixed **date_sunset** (int timestamp , int format , float latitude , float longitude , float zenith , float gmt_offset)

`date_sunset` retourne l'heure de coucher du soleil pour un jour (spécifié par le paramètre timestamp) et un endroit donnés. Les paramètres latitude , longitude et zenith ont pour valeurs par défaut, respectivement, les valeurs des options de configuration `date.default_latitude` , `date.default_longitude` et `date.sunrise_zenith` .

La valeur par défaut du paramètre latitude est North . Si vous voulez passer une valeur correspondant au Sud, vous devez donc passer une valeur négative. De même pour le paramètre longitude , qui, par défaut, vaut East .

Le paramètre `gmt_offset` est spécifié en heures.

constante	description	exemple
SUNFUNCS_RET_STRING		16:46

	retourne le résultat sous forme de <u>chaîne de caractères</u>	
SUNFUNCS_RET_DOUBLE	retourne le résultat sous la forme d'un <u>nombre à virgule flottante</u>	16.78243132
SUNFUNCS_RET_TIMESTAMP	retourne le résultat sous la forme d'un <u>entier</u> (timestamp)	1095034606

Exemple avec date_sunset

```
<?php
/* calcul de l'heure de coucher du soleil pour la ville de Lisbonne, Portugal
Latitude: 38.4 North
Longitude: 9 West
Zenith ~= 90
offset: +1 GMT
*/
echo 'Heure de coucher du soleil pour le ' . date("d/m/Y"). ' : ' .date_sunset(time(), SUNFUNCS_R
?>
```

L'exemple ci-dessus va afficher quelque chose de similaire à :

Heure de coucher du soleil pour le 20/12/2004 : 18:13

Voir aussi date_sunrise .

8.20.13 date() : Formate une date/heure locale

string **date** (string format , int timestamp)

date retourne une date sous forme d'une chaîne, au format donné par la chaîne format . La date est fournie par le paramètre timestamp , sous la forme d'un timestamp. Par défaut, la date courante est utilisée.

Array Array

Note
L'intervalle de validité d'un timestamp va généralement du Vendredi 13 Décembre 1901 20:45:54 GMT au Mardi 19 Janvier 2038 03:14:07 GMT. (Ces dates correspondent aux valeurs minimales et maximales des entiers 32 bits non-signés). Sur les systèmes Windows, cette intervalle va du 01-01-1970 au 19-01-2038.
Note
Pour générer un timestamp à partir d'une représentation de date, vous pouvez utiliser la fonction <u>strtotime</u> . De plus, certaines bases de données disposent de fonctions pour convertir leurs propres formats de date en timestamps (par exemple, MySQL et sa fonction <u>UNIX_TIMESTAMP()</u>).

Les caractères suivants sont utilisés pour spécifier le format :

Caractère	Description	Exemple de valeur
Jour	---	---
d	Jour du mois, sur deux chiffres (avec un zéro initial)	01 à 31
D	Jour de la semaine, en trois lettres (et en anglais)	Mon à Sun

8.20.12 date_sunset() : Retourne l'heure de coucher du soleil pour unjour et un endroit donnés 461

j	Jour du mois sans les zéros initiaux	1 à 31
l (L minuscule)	Jour de la semaine, textuel, version longue, en anglais	Sunday à Saturday
N	Représentation numérique ISO-8601 du jour de la semaine (ajouté en PHP 5.1.0)	1 (pour Lundi) à 7 (pour Dimanche)
S	Suffixe ordinal d'un nombre pour le jour du mois, en anglais, sur deux lettres	st , nd , rd ou th . Fonctionne bien avec j
w	Jour de la semaine au format numérique	0 (pour dimanche) à 6 (pour samedi)
z	Jour de l'année	0 à 366
Semaine	---	---
W	Numéro de semaine dans l'année ISO-8601, les semaines commencent le lundi (ajouté en PHP 4.1.0)	Exemple : 42 (la 42ème semaine de l'année)
Mois	---	---
F	Mois, textuel, version longue; en anglais, comme January ou December	January à December
m	Mois au format numérique, avec zéros initiaux	01 à 12
M	Mois, en trois lettres, en anglais	Jan à Dec
n	Mois sans les zéros initiaux	1 à 12
t	Nombre de jours dans le mois	28 à 31
Année	---	---
L	Est ce que l'année est bissextile	1 si bissextile, 0 sinon.
o	L'année ISO-8601. C'est la même valeur que Y , excepté que si le numéro de la semaine ISO (W) appartient à l'année précédente ou suivante, cette année sera utilisé à la place. (ajouté en PHP 5.1.0)	Exemples : 1999 ou 2003
Y	Année sur 4 chiffres	Exemples : 1999 et 2003
y	Année sur 2 chiffres	Exemples : 99 et 03
Heure	---	---
a	Ante meridiem et Post meridiem en minuscules	am ou pm
A	Ante meridiem et Post meridiem en majuscules	AM ou PM
B	Heure Internet Swatch	000 à 999
g	Heure, au format 12h, sans les zéros initiaux	1 à 12
G	Heure, au format 24h, sans les zéros initiaux	0 à 23
h	Heure, au format 12h, avec les zéros initiaux	01 à 12
H	Heure, au format 24h, avec les zéros initiaux	00 à 23
i	Minutes avec les zéros initiaux	00 à 59
s	Secondes, avec zéros initiaux	00 à 59
Fuseau horaire	---	---
e	L'identifiant du fuseau horaire (ajouté en PHP 5.1.0)	Exemple s: UTC , GMT , Atlantic/Azores
l (i majuscule)	L'heure d'été est activée ou pas	1 si oui, 0 sinon.
O	Différence d'heures avec l'heure de Greenwich (GMT), exprimée en heures	Exemple : +0200
T	Fuseau horaire du serveur	Exemples : EST , MDT ...

Z	Décalage horaire en secondes. Le décalage des zones à l'ouest de la zone UTC est négative, et à l'est, il est positif.	-43200 à 43200
Date et Heure complète	---	---
c	Date au format ISO 8601 (ajouté en PHP 5)	2004-02-12T15:19:21+00:00
r	Format de date RFC 2822	Exemple : Thu, 21 Dec 2000 16:01:07 +0200
U	Secondes depuis l'époque Unix (1er Janvier 1970, 0h00 00s GMT)	Voir aussi time

Les caractères non reconnus seront imprimés tels quel. " Z " retournera toujours "0" lorsqu'il est utilisé avec [gmdate](#) .

<p>Exemple avec date</p> <pre><?php // Définit le fuseau horaire par défaut à utiliser. Disponible depuis PHP 5.1 date_default_timezone_set('UTC'); // Affichage de quelque chose comme : Monday echo date("l"); // Affichage de quelque chose comme : Monday 15th of August 2005 03:12:46 PM echo date('l dS \of F Y h:i:s A'); // Affiche : July 1, 2000 is on a Saturday echo "July 1, 2000 is on a " . date("l", mktime(0, 0, 0, 7, 1, 2000)); /* utilise les constantes dans le paramètre format */ // Affichage de quelque chose comme : Mon, 15 Aug 2005 15:12:46 UTC echo date(DATE_RFC822); // Affichage de quelque chose comme : 2000-07-01T00:00:00+0000 echo date(DATE_ATOM, mktime(0, 0, 0, 7, 1, 2000)); ?></pre>

Vous pouvez faire afficher un caractère spécial dans la chaîne de format en le protégeant par un anti-slash. Si le caractère est lui-même une séquence incluant un anti-slash, vous devrez protéger aussi l'anti-slash.

<p>Protection des caractères avec date</p> <pre><?php // Affichage de quelque chose comme : Wednesday the 15th echo date("l \\t\\h\\e jS"); ?></pre>

Il est possible d'utiliser [date](#) et [mktime](#) ensemble pour générer des dates dans le futur ou dans le passé.

<p>Exemples avec date et mktime</p> <pre><?php \$demain = mktime(0, 0, 0, date("m") , date("d") + 1, date("Y")); \$le_mois_dernier = mktime(0, 0, 0, date("m")-1, date("d"), date("Y")); \$l_an_prochain = mktime(0, 0, 0, date("m"), date("d"), date("Y") + 1); ?></pre>
<p>Note</p>

Cette méthode est plus sûre que simplement ajouter ou retrancher le nombre de secondes dans une journée ou un mois à un timestamp, à cause des heures d'hiver et d'été.

Voici maintenant quelques exemples de formatage avec `date`. Notez que vous devriez échapper tous les autres caractères, car s'ils ont une signification spéciale, ils risquent de produire des effets secondaires indésirables. Notez aussi que les versions futures de PHP peuvent attribuer une signification à des lettres qui sont actuellement inertes. Lorsque vous échappez les caractères, pensez à utiliser des guillemets simples, pour que les séquences `\n` ne deviennent pas des nouvelles lignes.

Formatage avec `date`

```
<?php
// Aujourd'hui, le 12 Mars 2001, 10:16:18 pm

$aujourd'hui = date("F j, Y, g:i a");           // March 12, 2001, 10:16 pm
$aujourd'hui = date("m.d.y");                 // 03.12.01
$aujourd'hui = date("j, m, Y");               // 12, 3, 2001
$aujourd'hui = date("Ymd");                   // 20010312
$aujourd'hui = date('h-i-s, j-m-y, it is w Day z '); // 05-16-17, 12-03-01, 1631 1618 6 Monpm01
$aujourd'hui = date('\C'\e\s\t\ \l\e\ jS \j\o\u\r\.'); // C'est le 12th jour.
$aujourd'hui = date("D M j G:i:s T Y");       // Mon Mar 12 15:16:08 MST 2001
$aujourd'hui = date('H:m:s \m \e\s\t\ \l\e\ \m\o\i\s'); // 17:03:18 m est le mois
$aujourd'hui = date("H:i:s");                 // 10:16:18

// notation française

$aujourd'hui = date("d/m/y");                 // 12/03/01
$aujourd'hui = date("d/m/Y");                 // 12/03/2001
?>
```

Pour formater des dates dans d'autres langues, utilisez les fonctions `setlocale` et `strftime` au lieu de la fonction `date`.

Voir aussi `getlastmod`, `time`, `strftime`, `gmdate` et `mktime`.

8.20.14 `getdate()` : Retourne la date/heure

array `getdate` (*int timestamp*)

`getdate` retourne un tableau associatif contenant les informations de date et d'heure du timestamp `timestamp` (lorsqu'il est fourni, sinon, le timestamp de la date/heure courante), avec les champs suivants :

Clé	Description	Exemple de valeur retournée
"seconds"	Représentation numérique des secondes	0 à 59
"minutes"	Représentation numérique des minutes	0 à 59
"hours"	Représentation numérique des heures	0 à 23
"mday"	Représentation numérique du jour du mois courant	1 à 31
"wday"	Représentation numérique du jour de la semaine courante	0 (pour Dimanche) à 6 (pour Samedi)
"mon"	Représentation numérique du mois	1 à 12
"year"	Année, sur 4 chiffres	Exemples: 1999 ou 2003
"yday"	Représentation numérique du jour de l'année	0 à 365
"weekday"	Version texte du jour de la semaine	Sunday à Saturday
"month"	Version texte du mois, comme January ou March	January à December

0	Nombre de secondes depuis l'époque Unix, similaire à la valeur retournée par la fonction <code>time</code> et utilisée par <code>date</code> .	Dépend du système, typiquement de -2147483648 à 2147483647 .
---	--	--

Exemple avec `getdate`

```
<?php
$today = getdate();
print_r($today);
?>
```

L'exemple ci-dessus va afficher quelque chose de similaire à :

```
Array
(
[seconds] => 40
[minutes] => 58
[hours] => 21
[mday] => 17
[wday] => 2
[mon] => 6
[year] => 2003
[yday] => 167
[weekday] => Tuesday
[month] => June
[0] => 1055901520
)
```

Voir aussi `date` , `time` et `setlocale` .

8.20.15 `gettimeofday()` : Retourne l'heure actuelle

mixed **`gettimeofday`** (*bool* *return_float*)

`gettimeofday` est une interface vers `gettimeofday(2)`. Elle retourne un tableau associatif qui contient les informations retournées par le système.

Depuis PHP 5.1.0, il y a un paramètre optionnel, `return_float` , qui fait que `gettimeofday` retourne un nombre à virgule flottante lorsqu'il est défini à TRUE .

Clés du tableau :

- "sec" - secondes depuis l'époque Unix
- "usec" - microsecondes
- "minuteswest" - minutes de décalage par rapport à Greenwich, vers l'Ouest.
- "dsttime" - type de correction dst

Exemple avec `gettimeofday`

```
<?php
print_r(gettimeofday());

echo gettimeofday(true);
?>
```

L'exemple ci-dessus va afficher quelque chose de similaire à :

```
Array
(
    [sec] => 1073504408
    [usec] => 238215
    [minuteswest] => 0
    [dsttime] => 1
)
1073504408.23910
```

8.20.16 gmdate() : Formate une date/heure GMT/CUT

string **gmdate** (string format , int timestamp)

gmdate est identique à la fonction **date** , hormis le fait que le temps retourné est GMT (Greenwich Mean Time). Par exemple, en Finlande (GMT +0200), la première ligne ci-dessous affiche "Jan 01 1998 00:00:00", tandis que la seconde "Dec 31 1997 22:00:00".

Exemple avec gmdate

```
<?php
echo date("M d Y H:i:s", mktime(0, 0, 0, 1, 1, 1998));
echo gmdate("M d Y H:i:s", mktime(0, 0, 0, 1, 1, 1998));
?>
```

Note

Avant PHP 5.1.0, les timestamps négatifs (les dates avant 1970) ne fonctionnent pas sous certains systèmes (Windows, par exemple).

Voir aussi [date](#) , [strftime](#) , [mktime](#) et [gmmktime](#) .

8.20.17 gmmktime() : Retourne le timestamp UNIX d'une date GMT

int **gmmktime** (int hour , int minute , int second , int month , int day , int year , int is_dst)

gmmktime est identique à [mktime](#) hormis le fait que les paramètres passés sont GMT.

Les paramètres représentent toujours une date GMT donc, le paramètre `is_dst` n'influence pas le résultat.

Comme [mktime](#) , les arguments restants peuvent être ignorés. Ils prendront alors leur valeur GMT actuelle.

Note

Depuis PHP 5.1.0, le paramètre `is_dst` est devenu déprécié. Comme résultat, le nouveau gestionnaire de fuseau horaire doit être utilisé à la place.

Note

La fonction [gmmktime](#) utilise en interne la fonction [mktime](#) , donc, seuls les temps valides dans la zone locale dérivée peuvent être utilisés.

Exemple avec [gmmktime](#) à la frontière

```
<?php
gmmktime(0, 0, 0, 1, 1, 1970); // valide en GMT et l'ouest, invalide à l'est
?>
```

Voir aussi [mktime](#) , [date](#) et [time](#) .

8.20.18 gmstrftime() : Formate une date/heure GMT/CUT en fonction des paramètres locaux

string **gmstrftime** (string format , int timestamp)

[gmstrftime](#) se comporte exactement comme [strftime](#) hormis le fait que l'heure utilisée est celle de Greenwich (Greenwich Mean Time (GMT)). Par exemple, dans la zone Eastern Standard Time (est des USA) (GMT -0500), la première ligne de l'exemple ci-dessous affiche "Dec 31 1998 20:00:00", tandis que la seconde affiche "Jan 01 1999 01:00:00".

Exemple avec gmstrftime

```
<?php
setlocale(LC_TIME, 'en_US');
echo strftime("%b %d %Y %H:%M:%S", mktime (20, 0, 0, 12, 31, 98))."\n";
echo gmstrftime("%b %d %Y %H:%M:%S", mktime (20, 0, 0, 12, 31, 98))."\n";
?>
```

Voir aussi [strftime](#) .

8.20.19 idate() : Formate une date/heure locale en tant qu'entier

int **idate** (string format , int timestamp)

[idate](#) retourne un nombre formaté avec le format format et représentant le timestamp timestamp ou l'heure courant si timestamp est omis. En d'autres termes, le paramètre timestamp est optionnel et la valeur par défaut est la valeur retournée par la fonction [time](#) .

A l'inverse de la fonction [date](#) , [idate](#) accepte juste un caractère comme paramètre format .

Caractères de format	Description
B	Temps Internet Swatch Beat
d	Le jour du mois
h	Heure (format 12 heures)
H	Heure (format 24 heures)
i	Minutes
l	Retourne 1 si l'heure d'été est activée, 0 sinon
L	Retourne 1 pour une année bissextile, 0 sinon
m	Numéro du mois
s	Secondes
t	Jour du mois courant
U	Secondes depuis l'époque Unix - 1 Janvier 1970 00:00:00 GMT - c'est la même chose que la fonction time
w	Jour de la semaine (0 pour Dimanche)

W	Le numéro de semaine de l'année ; selon l'ISO-8601, les semaines débutent le Lundi
y	Année sur 1 ou 2 chiffres, voir la note plus bas
Y	Année sur 4 chiffres
z	Jour de l'année
Z	Décalage horaire en secondes
Note	
Sachant que <code>idate</code> retourne toujours un entier et qu'il ne peut commencer par 0 , <code>idate</code> peut retourner moins de chiffres que ce dont on pourrait espérer. Voir l'exemple ci-dessous :	

Exemple avec <code>idate</code> avec comme retour, un entier d'un seul chiffre
<pre><?php \$timestamp = strtotime('1st January 2004'); //1072915200 // ceci affiche l'année sur deux chiffres // néanmoins, vu que ce chiffre va commencer par "0", // seul "4" sera affiché echo idate('y', \$timestamp); ?></pre>

Voir aussi `date` et `time` .

8.20.20 localtime() : Lit l'heure locale

array **localtime** (int *timestamp* , bool *is_associative*)

`localtime` retourne un tableau identique à la structure retournée par la fonction C `localtime`. Le premier argument `timestamp` est un timestamp UNIX. S'il n'est pas fourni, l'heure courante est utilisée. Le second argument `is_associative` , s'il est mis à FALSE ou ignoré, force `localtime` à retourner un tableau à index numérique. S'il est mis à TRUE , `localtime` retourne un tableau associatif, avec tous les éléments de la structure C, accessible avec les clés suivantes :

- "tm_sec" - secondes
- "tm_min" - minutes
- "tm_hour" - heure
- "tm_mday" - jour du mois
- "tm_mon" - mois de l'année, commence par 0 pour Janvier
- "tm_year" - Année, incompatible an 2000
- "tm_wday" - Jour de la semaine
- "tm_yday" - Jour de l'année
- "tm_isdst" - Est-ce que l'heure d'hiver a pris effet?

Note
Les mois vont de 0 (Janvier) à 11 (Décembre) et les jours, de 0 (Dimanche) à 6 (Samedi).

Exemple avec <code>time</code>
<pre><?php \$localtime = localtime(); \$localtime_assoc = localtime(time(), 1); print_r(\$localtime); print_r(\$localtime_assoc); ?></pre>

L'exemple ci-dessus va afficher quelque chose de similaire à :

```
Array
(
[0] => 24
[1] => 3
[2] => 19
[3] => 3
[4] => 3
[5] => 105
[6] => 0
[7] => 92
[9] => 1
)

Array
(
[tm_sec] => 24
[tm_min] => 3
[tm_hour] => 19
[tm_mday] => 3
[tm_mon] => 3
[tm_year] => 105
[tm_wday] => 0
[tm_yday] => 92
[tm_isdst] => 1
)
```

8.20.21 `microtime()` : Retourne le timestamp UNIX actuel avec microsecondes

mixed `microtime` (*bool* `get_as_float`)

`microtime` retourne le timestamp Unix avec les microsecondes. Cette fonction est uniquement disponible sur les systèmes qui supportent la fonction `gettimeofday()`.

Lorsque cette fonction est appelée sans l'argument optionnel, elle retourne la chaîne "msec sec" avec sec qui est mesurée en secondes depuis le début de l'époque UNIX, (1er janvier 1970 00:00:00 GMT), et msec qui est le nombre de microsecondes de cette heure. Cette fonction est seulement disponible sur les systèmes d'exploitation qui supportent la fonction système `gettimeofday()`.

Si `get_as_float` est fourni et vaut TRUE , `microtime` retourne un nombre à virgule.

Note

Le paramètre `get_as_float` a été ajouté en PHP 5.0.0.

Exemple avec `microtime`

```
<?php
/**
 * Fonction simple identique à celle en PHP 5 qui va suivre
 */
function microtime_float()
```

```

{
    list($usec, $sec) = explode(" ", microtime());
    return ((float)$usec + (float)$sec);
}

$time_start = microtime_float();

// Attend pendant un moment
usleep(100);

$time_end = microtime_float();
$time = $time_end - $time_start;

echo "Ne rien faire pendant $time secondes\n";
?>

```

Exemple avec microtime en PHP 5

```

<?php
$time_start = microtime(true);

// Attend pendant un moment
usleep(100);

$time_end = microtime(true);
$time = $time_end - $time_start;

echo "Re rien faire pendant $time secondes\n";
?>

```

Voir aussi [time](#) .

8.20.22 mktime() : Retourne le timestamp UNIX d'une date

`int mktime (int hour , int minute , int second , int month , int day , int year , int is_dst)`

mktime retourne un timestamp UNIX correspondant aux arguments fournis. Ce timestamp est un entier long, contenant le nombre de secondes entre le début de l'époque UNIX (1er Janvier 1970 00:00:00 GMT) et le temps spécifié.

Les arguments peuvent être omis, de droite à gauche, et tous les arguments manquants sont utilisés avec la valeur courante de l'heure et du jour.

8.20.23 strftime() : Formate une date/heure locale avec les options locales

`string strftime (string format , int timestamp)`

strftime retourne la date sous la forme d'une chaîne formatée conformément au format `format` , en utilisant le timestamp `timestamp` donné. Si le timestamp est omis, la date actuelle est utilisée. Les mois et jours de la semaine, et toutes les chaînes dépendantes de la langue sont fixées avec la commande [setlocale](#) .

Les caractères suivants sont utilisés pour spécifier le format de la date :

- %a : nom abrégé du jour de la semaine (local).
- %A : nom complet du jour de la semaine (local).

- %b : nom abrégé du mois (local).
- %B : nom complet du mois (local).
- %c : représentation préférée pour les dates et heures, en local.
- %C : numéro de siècle (l'année, divisée par 100 et arrondie entre 00 et 99)
- %d : jour du mois en numérique (intervalle 01 à 31)
- %D : identique à %m/%d/%y
- %e : numéro du jour du mois. Les chiffres sont précédés d'un espace (de ' 1' à '31')
- %g : identique à %G, sur 2 chiffres.
- %G : L'année sur 4 chiffres correspondant au numéro de semaine (voir %V). Même format et valeur que %Y, excepté que si le numéro de la semaine appartient à l'année précédente ou suivante, l'année courante sera utilisé à la place.
- %h : identique à %b
- %H : heure de la journée en numérique, et sur 24-heures (intervalle de 00 à 23)
- %I : heure de la journée en numérique, et sur 12- heures (intervalle 01 à 12)
- %j : jour de l'année, en numérique (intervalle 001 à 366)
- %m : mois en numérique (intervalle 1 à 12)
- %M : minute en numérique
- %n : newline character
- %p : soit 'am' ou 'pm' en fonction de l'heure absolue, ou en fonction des valeurs enregistrées en local.
- %r : l'heure au format a.m. et p.m.
- %R : l'heure au format 24h
- %S : secondes en numérique
- %t : tabulation
- %T : l'heure actuelle (égal à %H:%M:%S)
- %u : le numéro de jour dans la semaine, de 1 à 7. (1 représente Lundi)

Attention

Les systèmes Sun Solaris semblent commencer le Dimanche à 1 malgré le fait que le standard ISO 9889:1999 (actuellement le standard C) spécifie clairement le début de la semaine à Lundi.

- %U : numéro de semaine dans l'année, en considérant le premier dimanche de l'année comme le premier jour de la première semaine.
- %V : le numéro de semaine comme défini dans l'ISO 8601:1988, sous forme décimale, de 01 à 53. La semaine 1 est la première semaine qui a plus de 4 jours dans l'année courante, et dont Lundi est le premier jour. (Utilisez %G ou %g pour les éléments de l'année qui correspondent au numéro de la semaine pour le timestamp donné.)
- %W : numéro de semaine dans l'année, en considérant le premier lundi de l'année comme le premier jour de la première semaine
- %w : jour de la semaine, numérique, avec Dimanche = 0
- %x : format préféré de représentation de la date sans l'heure
- %X : format préféré de représentation de l'heure sans la date
- %y : l'année, numérique, sur deux chiffres (de 00 à 99)
- %Y : l'année, numérique, sur quatre chiffres
- %Z ou %z : fuseau horaire, ou nom ou abréviation
- %% : un caractère '%' littéral

Note

Tous les caractères suivants ne sont pas toujours supportés par toutes les bibliothèques C. Dans ce cas, ils ne seront pas supportés par PHP non plus. De plus, toutes les plates-formes ne supportent pas les timestamps négatifs, et vos dates pourraient être limitées par le début de l'époque Unix. Cela signifie que %e, %T, %R et %D (et peut être d'autres) et les dates antérieures au 1er Janvier 1970 ne fonctionneront pas sous Windows, sur certaines distributions de Linux, et sur certains systèmes d'exploitation. Pour Windows, une liste complète des options de conversion est disponible sur le [site de MSDN](#).

Exemple avec `strftime`

```
<?php
setlocale(LC_TIME, "C");
echo strftime("%A en Finlandais est ");
setlocale(LC_TIME, "fi");
echo strftime("%A, en Français ");
setlocale(LC_TIME, "fr");
echo strftime("%A et en Allemand ");
setlocale(LC_TIME, "de");
echo strftime("%A.\n");
?>
```

Cet exemple ne fonctionnera que si vous avez les locales respectives installées sur votre système.

Note

%G et %V, qui sont basées sur la semaine ISO 8601:1988, peut conduire à des résultats inattendus (bien que corrects) si le système de numérotation n'est pas connu. Voyez l'exemple %V ci-dessus et l'exemple ci-dessous.

Exemple au format de date ISO 8601:1988

```
<?php
/*      December 2002 / January 2003
ISOWk  M   Tu  W   Thu F   Sa  Su
-----
51     16 17 18  19 20  21  22
52     23 24 25  26 27  28  29
1      30 31  1  2   3   4   5
2       6  7  8   9 10  11  12
3       13 14 15 16 17  18  19  */

// Affiche : 12/28/2002 - %V,%G,%Y = 52,2002,2002
echo "12/28/2002 - %V,%G,%Y = " . strftime("%V,%G,%Y",strtotime("12/28/2002")) . "\n";

// Affiche : 12/30/2002 - %V,%G,%Y = 1,2003,2002
echo "12/30/2002 - %V,%G,%Y = " . strftime("%V,%G,%Y",strtotime("12/30/2002")) . "\n";

// Affiche : 1/3/2003 - %V,%G,%Y = 1,2003,2003
echo "1/3/2003 - %V,%G,%Y = " . strftime("%V,%G,%Y",strtotime("1/3/2003")) . "\n";

// Affiche : 1/10/2003 - %V,%G,%Y = 2,2003,2003
echo "1/10/2003 - %V,%G,%Y = " . strftime("%V,%G,%Y",strtotime("1/10/2003")) . "\n";

/*      December 2004 / January 2005
ISOWk  M   Tu  W   Thu F   Sa  Su
-----
51     13 14 15  16 17  18  19
52     20 21 22  23 24  25  26
53     27 28 29  30 31  1   2
1       3  4  5   6  7   8   9
2       10 11 12 13 14  15  16  */

// Affiche : 12/23/2004 - %V,%G,%Y = 52,2004,2004
echo "12/23/2004 - %V,%G,%Y = " . strftime("%V,%G,%Y",strtotime("12/23/2004")) . "\n";

// Affiche : 12/31/2004 - %V,%G,%Y = 53,2004,2004
echo "12/31/2004 - %V,%G,%Y = " . strftime("%V,%G,%Y",strtotime("12/31/2004")) . "\n";

// Affiche : 1/2/2005 - %V,%G,%Y = 53,2004,2005
echo "1/2/2005 - %V,%G,%Y = " . strftime("%V,%G,%Y",strtotime("1/2/2005")) . "\n";

// Affiche : 1/3/2005 - %V,%G,%Y = 1,2005,2005
echo "1/3/2005 - %V,%G,%Y = " . strftime("%V,%G,%Y",strtotime("1/3/2005")) . "\n";
?>
```

Voir aussi [setlocale](#) , [mktime](#) , [strptime](#) et le [groupe de spécifications de strftime](#) .

8.20.24 [strptime\(\)](#) : Analyse une date générée par [strftime](#)

array [strptime](#) (string date , string format)

[strptime](#) retourne un tableau après avoir analysé date , ou FALSE en cas d'erreur.

Les noms des mois et jours de la semaine dépendent de la configuration locale, choisie avec [setlocale](#) (LC_TIME).

Note

Cette fonction n'est pas implémentée sous Windows.

8.20.25 [strtotime\(\)](#) : Transforme un texte anglais en timestamp

int [strtotime](#) (string time , int now)

[strtotime](#) essaye de lire une date au format anglais US dans la chaîne time , et de la transformer en timestamp Unix (le nombre de secondes depuis le 1er Janvier 1970 à 00:00:00 GMT), relativement au timestamp now , ou à la date courante si ce dernier est omis.

Cette fonction devrait utiliser la variable d'environnement **TZ** (si disponible) pour calculer le timestamp. Depuis PHP 5.1.0, il y a une façon simple de définir un fuseau horaire à utiliser avec toutes les fonctions de date/heure. Le processus est expliqué dans la page du manuel de la fonction [date_default_timezone_get](#) .

Note

Si l'année est spécifiée sur deux chiffres, les valeurs entre 0 et 69 correspondent aux années entre 2000 et 2069 et les valeurs entre 70 et 100 correspondent aux années entre 1970 et 2000.

8.20.26 [time\(\)](#) : Retourne le timestamp UNIX actuel

int [time](#) (void)

[time](#) retourne l'heure courante, mesurée en secondes depuis le début de l'époque UNIX, (1er janvier 1970 00:00:00 GMT).

Array

Exemple avec [time](#)

```
<?php
$nextWeek = time() + (7 * 24 * 60 * 60);
// 7 jours; 24 heures; 60 minutes; 60secondes
echo 'Aujourd\'hui :      '. date('Y-m-d') ."\n";
echo 'Semaine prochaine : '. date('Y-m-d', $nextWeek) ."\n";
?>
```

L'exemple ci-dessus va afficher quelque chose de similaire à :

Aujourd'hui : 2005-03-30

Semaine prochaine : 2005-04-07

Voir aussi [date](#) et [microtime](#) .

8.21 DBA

8.21.1 Introduction

Ces fonctions constituent l'interface avec les bases de type Berkeley.

C'est une couche générale pour plusieurs bases de données sur fichiers. En tant que telles, les fonctionnalités sont limitées à une partie des fonctionnalités des bases de données modernes, comme Sleepycat Software's DB2 (à ne pas confondre avec IBM's DB2 software, qui fonctionne avec ODBC).

8.21.2 Pré-requis

Le comportement de certaines fonctions dépend de la base de données utilisée. Par exemple dba_optimize et dba_sync n'auront pas le même effet d'une base à l'autre.

Lors de l'utilisation de la fonction dba_open ou de dba_popen , une des bibliothèques suivantes doit être fournie comme argument. La liste complète des bibliothèques supportées par votre configuration est disponible avec la fonction phpinfo . (Pour inclure le support de l'une de ces bibliothèques, ajoutez l'option de configuration `--with-XXXX`).

Bibliothèque	Notes
dbm	Dbm est la plus ancienne des base de données de type Berkeley. Il vaut mieux l'éviter, si possible. Les fonctions de compatibilités codées dans DB2 et gdbm ne sont pas supportées, car elles ne sont compatibles qu'au niveau du code source et ne peuvent pas gérer le format dbm originel.
ndbm	ndbm est un nouveau type de dbm plus flexible. Il a cependant la majorité des limitations du genre. (il est donc déconseillé)
gdbm	gdbm est <u>la base dbm GNU</u> .
db2	db2 est <u>DB2 de Sleepycat Software</u> . Elle se décrit comme un "ensemble d'outils qui fournissent une base de données performante, tant pour les applications indépendantes que pour le client/serveur".
db3	DB3 est le <u>DB3 de Sleepycat Software</u> .
db4	DB4 est le <u>Sleepycat Software's DB4</u> . Elle est disponible depuis PHP 4.3.2.
cdb	cdb est "un package rapide, robuste, léger pour créer et lire des bases de données constantes". C'est l'auteur de qmail qui l'a écrit, et elle est disponible <u>ici</u> . Puisque c'est une base constante, elle ne supporte que la lecture. Mais depuis PHP 4.3.0, le support de l'écriture a été ajouté (mais pas celui de la modification), grâce aux bibliothèques internes de cdb.
cdb_make	Depuis PHP 4.3.0, la création de fichier cdb est supportée, lorsque la bibliothèque cdb, distribuée avec PHP est utilisée.
flatfile	Depuis PHP 4.3.0, et pour assurer la compatibilité avec le gestionnaire <u>dbm</u> qui est obsolète et doit être évité, PHP supporte aussi un gestionnaire de fichiers. Il est recommandé de l'utiliser lorsque vous devez accéder à des fichiers au format dbm. Cela arrive lorsque vous ne pouvez pas trouver les bibliothèques dbm pour compiler PHP avec.
inifile	Ceci est disponible depuis PHP 4.3.3 pour pouvoir modifier les fichiers php.ini depuis les scripts PHP. Lorsque vous travaillez avec des fichiers ini, vous pouvez passer des tableaux de la forme <code>array(0 => groupe, 1=>nom_valeur)</code> ou des chaînes de la forme

	"[groupe]nom_valeur" où groupe est optionnel. Comme les fonctions dba_firstkey et dba_nextkey retournent des représentations de la clé au format chaîne, il existe une nouvelle fonction, dba_key_split , disponible depuis PHP 5 qui permet de convertir la chaîne de caractères de clés en tableau de clés sans perdre FALSE .
qdbm	Disponible depuis PHP 5.0.0. La bibliothèque qdbm peut être téléchargée sur http://qdbm.sourceforge.net .

Lors de l'utilisation de la fonction [dba_open](#) ou de [dba_popen](#), une des bibliothèques suivantes doit être fournie comme argument. La liste complète des bibliothèques supportées par votre configuration est disponible avec la fonction [phpinfo](#) .

8.21.3 Installation

En utilisant l'option de compilation `--enable-dba=shared`, vous pouvez compiler un module dynamique qui activera le support des bases de données de style DBM pour PHP. Vous devez aussi ajouter le support d'au moins l'un des gestionnaires suivants, en spécifiant l'option de configuration `--with-XXXX` lors de la configuration de PHP.

Attention	
Après avoir configuré et compilé PHP, vous devez exécuter les test suivants à partir de la ligne de commande : <code>php run-tests.php ext/dba</code> . Cela montre si votre combinaison de descripteurs fonctionne. Les plus problématiques sont dbm et ndbm qui entrent en conflit avec nombre d'installations. Ceci est dû au fait que sur de nombreux systèmes, ces bibliothèques font parties de plus d'une bibliothèque. Le test de configuration vous empêche juste de configurer des descripteurs dont la combinaison est défectueuse alors qu'ils fonctionnent correctement séparément.	

Gestionnaire	Option de configuration
dbm	Pour activer le support de dbm, ajoutez <code>--with-dbm[=DIR]</code> .
	Note dbm est une surcouche qui aboutit souvent à des échecs. Ainsi, vous ne devez utiliser dbm que si vous êtes sûrs qu'il fonctionne et que vous avez besoin de ce format.
ndbm	Pour activer le support de ndbm, ajoutez <code>--with-ndbm[=DIR]</code> .
	Note ndbm est une surcouche qui aboutit souvent à des échecs. Ainsi, vous ne devez utiliser ndbm que si vous êtes sûrs qu'il fonctionne et que vous avez besoin de ce format.
gdbm	Pour activer le support de gdbm, ajoutez <code>--with-gdbm[=DIR]</code> .
db2	Pour activer le support de db2, ajoutez <code>--with-db2[=DIR]</code> .
	Note db2 entre en conflit avec db3 et db4.
db3	Pour activer le support de db3, ajoutez <code>--with-db3[=DIR]</code> .
	Note db3 entre en conflit avec db2 et db4.
db4	Pour activer le support de db4, ajoutez <code>--with-db4[=DIR]</code> .
	Note db4 entre en conflit avec db2 et db3.
	Note

	Ceci a été introduit avec PHP 4.3.2. Dans des versions précédentes de PHP, vous devez utiliser <code>--with-db3=DIR</code> où DIR pointe vers le répertoire où se situe la bibliothèque db4. Il n'est pas possible d'utiliser des versions de db supérieures à 4.1 avec des versions de PHP antérieures à 4.3.0. De même, les bibliothèques db avec des versions allant de 4.1 à 4.1.24 ne peuvent être utilisées avec aucune version de PHP.
cdb	Pour activer le support de cdb, ajoutez <code>--with-cdb[=DIR]</code> .
	Note Depuis PHP 4.3.0, vous pouvez omettre d'utiliser DIR, afin d'exploiter la bibliothèque cdb fournie avec PHP, qui ajoute un gestionnaire <code>cdb_make</code> , permet la création de fichier cdb et permet l'accès aux fichiers cdb via le réseau avec les flux de PHP.
flatfile	Pour activer le support des fichiers, ajoutez <code>--with-flatfile</code> .
	Note Ceci a été ajouté à PHP 4.3.0 pour assurer la compatibilité avec l'extension <code>dbm</code> qui est obsolète. Utilisez ce gestionnaire seulement quand vous ne pouvez installer aucun autre gestionnaire et que vous ne pouvez pas utiliser le gestionnaire cdb intégré.
inifile	Pour activer le support de inifile , ajoutez l'option de compilation <code>--with-inifile</code> .
	Note Cette option a été ajoutée avec PHP 5.0.0 et permet de lire et écrire dans des fichiers d'initialisation de type Microsoft (<code>.ini</code>), comme le <code>php.ini</code> par exemple.
qdbm	Pour activer le support de qdbm, ajoutez l'option de compilation <code>--with-qdbm[=DIR]</code> .
	Note
	qdbm entre en conflit avec dbm et gdbm.
	Note Cette option a été ajoutée avec PHP 5.0.0. La bibliothèque qdbm peut être téléchargée sur http://qdbm.sourceforge.net .
Note	Jusqu'en PHP 4.3.0, il était possible d'utiliser simultanément les gestionnaires db2 et db3 mais juste avec un seul en interne. Cela signifie que vous ne pouvez pas avoir les deux formats de fichiers. Depuis PHP 5.0.0, une vérification est faite lors de la configuration pour éviter ces problèmes.

8.21.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.21.5 Types de ressources

Les fonctions `dba_open` et `dba_popen` retourne une ressource qui représente la base de données ouverte, et qui est utilisées par toutes les autres fonctions dba.

8.21.6 Constantes pré-définies

Cette extension ne définit aucune constante.

8.21.7 Exemples

Exemple DBA (1)

```
<?php
$cid = dba_open ("/tmp/test.db", "n", "db2");

if (!$cid) {
    echo "dba_open a échoué\n";
    exit;
}

dba_replace ("cle", "Ceci est un exemple!", $cid);

if (dba_exists ("cle", $cid)) {
    echo dba_fetch ("cle", $cid);
    dba_delete ("cle", $cid);
}

dba_close ($cid);
?>
```

DBA gère les données binaires et n'a pas de limites arbitraires. Elle hérite de toutes les limites de la base sous-jacente.

Toutes les bases de données sur fichiers doivent fournir un moyen de changer le mode d'accès au fichier d'une base et, si possible, de toutes les bases. Le mode d'accès est généralement passé en quatrième argument à [dba_open](#) ou [dba_popen](#) .

Vous pouvez accéder à toutes les entrées d'une base d'une manière linéaire, avec les fonctions [dba_firstkey](#) et [dba_nextkey](#) . Vous ne devez pas modifier une base lorsque vous la traversez ainsi.

Passer en revue une base

```
<?php
// ouverture de la base

$key = dba_firstkey($cid);

while ($key != false) {
    if ($quelquechose) {
        // conserver la clé pour faire d'autres opérations plus tard
        $handle_later[] = $key;
    }
    $key = dba_nextkey($cid);
}

foreach ($handle_later as $val) {
    dba_delete($val, $cid);
}

?>
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)

- [Exemples](#)
- [dba_close](#)
- [dba_delete](#)
- [dba_exists](#)
- [dba_fetch](#)
- [dba_firstkey](#)
- [dba_handlers](#)
- [dba_insert](#)
- [dba_key_split](#)
- [dba_list](#)
- [dba_nextkey](#)
- [dba_open](#)
- [dba_optimize](#)
- [dba_popen](#)
- [dba_replace](#)
- [dba_sync](#)

8.21.9 [dba_delete\(\)](#) : Efface une ligne dans une base DBA

bool **dba_delete** (string key , resource handle)

[dba_delete](#) efface l'entrée spécifiée par la clé key , dans la base identifiée par handle .

8.21.10 [dba_exists\(\)](#) : Vérifie qu'une clé DBA existe

bool **dba_exists** (string key , resource handle)

[dba_exists](#) vérifie si la clé key existe dans la base identifiée par handle .

8.21.11 [dba_fetch\(\)](#) : Lit les données liées à une clé DBA

string **dba_fetch** (string key , resource handle) string **dba_fetch** (string key , int skip , resource handle)

[dba_fetch](#) lit les données spécifiées par la clé key dans la base identifiée par handle .

8.21.12 [dba_firstkey\(\)](#) : Lit la première clé DBA

string **dba_firstkey** (resource handle)

[dba_firstkey](#) retourne la première clé de la base de données spécifiée par handle et y place le pointeur interne de clé. Cela permettra de traverser la base.

8.21.13 [dba_handlers\(\)](#) : Liste les gestionnaires DBA disponibles

array **dba_handlers** (bool full_info)

[dba_handlers](#) retourne un tableau avec tous les gestionnaires supportés par cette extension.

8.21.14 **dba_insert()** : Insère une entrée DBA

bool **dba_insert** (string key , string value , resource handle)

dba_insert insère l'entrée décrite par la clé key et la valeur value dans la base spécifiée par handle .

8.21.15 **dba_key_split()** : Transforme une représentation de clé DBA par chaîne en une représentation par tableau

mixed **dba_key_split** (mixed key)

dba_key_split transforme une représentation de clé DBA par chaîne en une représentation par tableau.

8.21.16 **dba_list()** : Liste tous les fichiers de bases de données DBA ouverts

array **dba_list** (void)

dba_list liste tous les fichiers de bases de données DBA ouverts.

8.21.17 **dba_nextkey()** : Lit la clé DBA suivante

string **dba_nextkey** (resource handle)

dba_nextkey retourne la clé suivante, dans la base identifiée par handle et incrémente le pointeur de clé.

8.21.18 **dba_open()** : Ouvre une base de données DBA

resource **dba_open** (string path , string mode , string handler , mixed ...)

dba_open établit une connexion à la base repérée par path avec le mode mode et l'identifiant handler .

8.21.19 **dba_optimize()** : Optimise une base DBA

bool **dba_optimize** (resource handle)

dba_optimize optimise la base de données identifiée par handle .

8.21.20 **dba_popen()** : Ouvre une connexion persistante à une base de données DBA

resource **dba_popen** (string path , string mode , string handler , mixed ...)

dba_popen établit une connexion persistante à la base repérée par path avec le mode mode , en utilisant l'identifiant handler .

8.21.21 dba_replace() : Remplace ou insère une ligne DBA

bool **dba_replace** (string key , string value , resource handle)

dba_replace remplace ou insère une entrée, pour la clé key et avec la valeur value dans la base identifiée par handle .

8.21.22 dba_sync() : Synchronise une base de données DBA

bool **dba_sync** (resource handle)

dba_sync synchronise la base de données spécifiée par handle . Si accepté, cela va probablement lancer une opération de réécriture physique du fichier.

8.22 dBase

8.22.1 Introduction

Ces fonctions vous permettront d'accéder aux enregistrements d'une base au format dBase (.dbf).

Les fichiers dBase sont de simples fichiers séquentiels d'enregistrements de longueur fixe. Les enregistrements sont ajoutés à la fin du fichier et les enregistrements supprimés sont conservés jusqu'à l'appel de `dbase_pack` .

Les types disponibles des champs dBase sont :

Champ	Type dBase	Format	Informations additionnelles
M	Memo	n/a	Ce type n'est pas supporté par PHP ; ce type de champ sera donc ignoré
D	Date	YYYYMMDD	La longueur de ce champ est limitée à 8
N	Number	Un nombre	Vous devez déclarer une longueur et une précision (le nombre de chiffres après le point décimal).
C	String	Une chaîne de caractères	Vous devez déclarer une longueur. Lors de la récupération de données, la chaîne sera complétée par la droite par des espaces pour arriver à la longueur spécifiée.
L	Boolean	T ou Y pour TRUE , F ou N pour FALSE	Stocké et retourné en tant qu'entier (1 ou 0)

Attention

dBase ne permet pas l'utilisation d'index, de "memo fields" , ni le blocage de la base. Deux processus de serveurs web différents modifiant le même fichier dBase risquent de rendre votre base de données incohérente.

Nous vous recommandons de ne pas utiliser les fichiers dBase comme base de données de production. Choisissez n'importe quel serveur SQL à la place. [MySQL](#) et [Postgres](#) sont des choix classiques avec PHP. Le support de dBase ne se justifie ici que pour vous permettre d'importer et d'exporter des données depuis et vers votre base de données issues du web, car ce format de fichier est communément accepté par les feuilles et assistants personnes type Windows.

8.22.2 Installation

Pour activer la bibliothèque dBase fournie avec PHP, vous devez compiler PHP avec l'option `--enable-dbase` .

8.22.3 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.22.4 Types de ressources

Cette extension ne définit aucune ressource.

8.22.5 Exemples

La plupart des exemples dans cette documentation nécessite une base de données dBase. Nous utiliserons /tmp/test.dbf qui est créée dans l'exemple de la fonction [dbase_create](#) .

8.22.6 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Exemples](#)
- [Constantes pré-définies](#)
- [dbase_add_record](#)
- [dbase_close](#)
- [dbase_create](#)
- [dbase_delete_record](#)
- [dbase_get_header_info](#)
- [dbase_get_record_with_names](#)
- [dbase_get_record](#)
- [dbase_numfields](#)
- [dbase_numrecords](#)
- [dbase_open](#)
- [dbase_pack](#)
- [dbase_replace_record](#)

8.22.8 dbase_close() : Ferme une base dBase

bool **dbase_close** (int dbase_identifieur)

[dbase_close](#) ferme la base associée à dbase_identifieur .

8.22.9 dbase_create() : Crée une base de données dBase

int **dbase_create** (string filename , array fields)

[dbase_create](#) crée une base de données dBase dans le fichier filename , et avec les champs fields .

Note

Lorsque le [safe-mode](#) est activé, PHP vérifie si le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.

Note

Cette fonction est affectée par la directive de configuration <code>open_basedir</code> .

8.22.10 `dbase_delete_record()` : Efface un enregistrement dans une base dBase

bool `dbase_delete_record` (int `dbase_identifieur` , int `record_number`)

`dbase_delete_record` marque l'enregistrement `record` pour l'effacement, dans la base `dbase_identifieur` .

Note

Pour effacer réellement l'enregistrement de la base de données, vous devez appeler la fonction <code>dbase_pack</code> .
--

8.22.11 `dbase_get_header_info()` : Récupère des informations d'en-tête d'une base de données dBase

array `dbase_get_header_info` (int `dbase_identifieur`)

`dbase_get_header_info` retourne des informations sur la structure des colonnes de la base de données référencée par `dbase_identifieur` .

8.22.12 `dbase_get_record_with_names()` : Lit un enregistrement dans une base dBase, sous la forme d'un tableau associatif

array `dbase_get_record_with_names` (int `dbase_identifieur` , int `record_number`)

`dbase_identifieur` retourne les données de l'enregistrement `record` dans un tableau associatif.

8.22.13 `dbase_get_record()` : Lit un enregistrement dans une base dBase

array `dbase_get_record` (int `dbase_identifieur` , int `record_number`)

`dbase_get_record` retourne les données de l'enregistrement `record` dans un tableau.

8.22.14 `dbase_numfields()` : Compte le nombre de champs d'une base dBase

int `dbase_numfields` (int `dbase_identifieur`)

`dbase_numfields` retourne le nombre de champs (colonnes) de la base de données `dbase_identifieur` .

Note

Les numéros de champs sont numérotés de 0 à <code>dbase_numfields(\$db)-1</code> , tandis que les numéros

d'enregistrements sont numérotés de 1 à `dbase_numrecords($db)` .

8.22.15 `dbase_numrecords()` : Compte le nombre d'enregistrements dans une base dBase

int **`dbase_numrecords`** (int `dbase_identifieur`)

`dbase_numrecords` retourne le nombre d'enregistrements (lignes) dans la base `dbase_identifieur` .

Note

Les numéros de champ sont numérotés de 0 à `dbase_numfields($db)-1` , tandis que les numéros d'enregistrements sont numérotés de 1 à `dbase_numrecords($db)` .

8.22.16 `dbase_open()` : Ouvre une base dBase

int **`dbase_open`** (string `filename` , int `mode`)

`dbase_open` ouvre une base de données dBase avec un mode d'accès donné.

Note

Lorsque le `safe-mode` est activé, PHP vérifie si le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.

Note

Cette fonction est affectée par la directive de configuration `open_basedir` .

8.22.17 `dbase_pack()` : Compacte une base dBase

bool **`dbase_pack`** (int `dbase_identifieur`)

`dbase_pack` compacte la base de données `dbase_identifieur` (effacement définitif de tous les enregistrements marqués pour l'effacement, avec la fonction `dbase_delete_record`).

8.22.18 `dbase_replace_record()` : Remplace un enregistrement dans une base dBase

bool **`dbase_replace_record`** (int `dbase_identifieur` , array `record` , int `record_number`)

`dbase_replace_record` remplace les données associées à l'enregistrement `dbase_record_number` par les données enregistrées dans `record` , dans la base `dbase_identifieur` .

8.23 DBM (Obsolète)

8.23.1 Introduction

Ces fonctions vous permettent d'écrire des lignes dans une base de données de type dbm. Ce type de base (supporté par Berkeley DB, [GDBM](#), quelques bibliothèques systèmes, ou certaines bibliothèques du système d'exploitation) enregistre les paires clés/valeurs, (contrairement aux enregistrements par ligne, utilisés par les autres bases de données relationnelles).

Note

Toutefois, le support dbm est obsolète, et il est fortement recommandé d'utiliser la [couche d'abstraction de base dba](#).

8.23.2 Pré-requis

Pour utiliser ces fonctions, vous devez compiler PHP avec le support d'un gestionnaire. Reportez-vous à la [liste](#) de gestionnaires supportés.

8.23.3 Installation

Pour utiliser ces fonctions, vous devez compiler PHP avec le support dbm en utilisant l'option de configuration `--with-db`. De plus, vous devez assurer le [support](#) pour une sous base de données, ou bien utiliser quelques bibliothèques du système.

8.23.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.23.5 Types de ressources

La fonction [dbmopen](#) retourne une ressource de base de données, utilisée par toutes les autres fonctions dbm.

8.23.6 Constantes pré-définies

Cette extension ne définit aucune constante.

8.23.7 Exemples

Présentation de dbm

```
<?php
$dbm = dbmopen ("lastseen", "w");
if (dbmexists ($dbm, $userid)) {
    $last_seen = dbmfetch ($dbm, $userid);
} else {
    dbminsert ($dbm, $userid, time());
}
```

```
do_stuff();
dbmreplace ($dbm, $userid, time());
dbmclose ($dbm);
?>
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [dblist](#)
- [dbmclose](#)
- [dbmdelete](#)
- [dbmexists](#)
- [dbmfetch](#)
- [dbmfirstkey](#)
- [dbminsert](#)
- [dbmnextkey](#)
- [dbmopen](#)
- [dbmreplace](#)

8.23.9 dbmclose() : Ferme une base de données dbm

bool **dbmclose** (resource dbm_identifieur)

[dbmclose](#) déverrouille et ferme la base de données dbm_identifieur .

8.23.10 dbmdelete() : Efface une valeur

bool **dbmdelete** (resource dbm_identifieur , string key)

[dbmdelete](#) efface la valeur de la clé key , dans la base dbm_identifieur .

[dbmdelete](#) retourne FALSE si la clé n'existe pas dans cette base.

8.23.11 dbmexists() : Indique si une valeur existe

bool **dbmexists** (resource dbm_identifieur , string key)

[dbmexists](#) retourne TRUE s'il y a une valeur associée à la clé key .

8.23.12 dbmfetch() : Lit une valeur

string **dbmfetch** (resource dbm_identifieur , string key)

[dbmfetch](#) retourne la valeur associée à la clé key .

8.23.13 dbmfirstkey() : Lit la première clé

string **dbmfirstkey** (resource dbm_identifieur)

dbmfirstkey retourne la première clé de la base de données. Notez bien que les clés ne sont pas dans un ordre défini, étant donné que la table est construite comme un tableau associatif.

8.23.14 dbminsert() : Insère une valeur

int **dbminsert** (resource dbm_identifieur , string key , string valeur)

dbminsert ajoute la valeur valeur dans la base de données, avec la clé key .

dbminsert retourne -1 si la base a été ouverte en mode lecture seule, 0 si l'insertion a été réussie, et 1 si la clé key existe déjà. (Pour remplacer la valeur, utilisez dbmreplace .)

8.23.15 dbmnextkey() : Lit la clé suivante

string **dbmnextkey** (resource dbm_identifieur , string key)

dbmnextkey retourne la clé après la clé key . En appelant dbmfirstkey , puis successivement dbmnextkey , il est possible de passer en revue toute les paires clé/valeur de la base de données dbm. Par exemple :

Passer en revue une base de données d'une base dbm

```

<?php
    $cle = dbmfirstkey($dbm_id);
    while ($cle){
        echo "$cle = " . dbmfetch($dbm_id, $cle) . "\n";
        $cle = dbmnextkey($dbm_id, $cle);
    }
?>

```

8.23.16 dbmopen() : Ouvre une base de données dbm

resource **dbmopen** (string filename , string flags)

Le premier argument est le chemin absolu jusqu'au fichier dbm à ouvrir. Le deuxième argument est le mode d'ouverture du fichier, qui peut prendre les valeurs suivantes : "r", "n", "c" ou "w" qui représentent respectivement lecture seule, nouveau (ce qui implique lecture/écriture, et qui, probablement, va écraser une base existante), création (ce qui implique lecture/écriture, et qui, probablement, va écraser une base existante), et lecture/écriture.

dbmopen retourne un identifiant qui sera passé à toutes les autres fonctions dbm en cas de succès, ou FALSE en cas d'échec.

Si ndbm est utilisé, ndbm va créer les fichiers filename.dir et filename.pag . gdbm n'utilise qu'un fichier, tout comme les bibliothèques internes, et Berkeley db crée le fichier filename.db . Notez que PHP dispose de son propre système de verrouillage des fichiers, qui s'additionne à celui éventuellement utilisé par les bibliothèques. PHP n'efface jamais les fichiers .lck qu'il crée. Il les utilise comme inode fixe, sur lequel faire le verrouillage. Pour plus d'informations sur les fichiers

dbm, reportez-vous à vos pages de manuel Unix (man) , ou bien chargez [GDBM GNU](http://prep.ai.mit.edu/pub/gnu) .
<ftp://prep.ai.mit.edu/pub/gnu> .

Note

Lorsque le <u>safe-mode</u> est activé, PHP vérifie si le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.

8.23.17 dbmreplace() : Remplace une valeur

int **dbmreplace** (resource dbm_identifieur , string key , string value)

dbmreplace remplace la valeur courante par la valeur value pour la clé key , dans une base dbm_identifieur .

dbmreplace crée la clé, si elle n'existe pas dans la base.

8.24 Fonctions DB++

Attention

Ce module est **EXPERIMENTAL** . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez ce module à vos risques et périls.

8.24.1 Introduction

DB++, qui est programmé par une société allemande [Concept asa](#) , est un système de base de données relationnelles à haute performance, faible consommation de mémoire et de disque. Bien qu'il comprenne le langage SQL en tant qu'interface supplémentaire, ce n'est pas vraiment une base de données SQL, car elle fournit son propre langage SQL, qui est plus influencé par l'algèbre relationnelle que ne l'est SQL.

Concept asa a toujours supporté les langages Open Source, et DB++ dispose d'interfaces Perl et Tcl depuis longtemps. Il utilise aussi Tcl pour ses procédures stockées internes.

8.24.2 Pré-requis

Cette extension dépend d'une bibliothèque cliente externe : vous devez donc avoir le client DB++ installé sur le système pour pouvoir compiler cette extension dans PHP.

[Concept asa](#) fournit [des versions de démo de DB++](#) et la [documentation](#) pour Linux, ainsi que certaines autres versions UNIX. Il y a aussi une version pour Windows de DB++, mais cette extension ne la supporte pas (pour le moment).

8.24.3 Installation

Pour compiler cette extension vous-même, vous devez avoir les bibliothèques clients et les fichiers d'en-têtes installés sur le système (ils sont inclus dans la distribution de DB++ par défaut). Exécutez la commande configure avec l'option --with-dbplus pour inclure cette extension dans PHP.

configure recherche les bibliothèques clients et les fichiers d'en-têtes dans les dossiers suivants : /usr/dbplus , /usr/local/dbplus et /opt/dbplus . Si vous avez installé DB++ dans un autre dossier, indiquez le chemin de ce dossier dans l'option de configure comme ceci :
--with-dbplus=/votre/dossier/installation .

8.24.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.24.5 Types de ressources

8.24.5.1 dbplus_relation

La plupart des fonctions DB++ travaillent avec une ressource de type dbplus_relation : c'est un pointeur sur une relation, ou une relation, généré par une requête.

8.24.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

8.24.6.1 Codes d'erreurs de DB++

Constante PHP	Constante DB++	Description
DBPLUS_ERR_NOERR (entier)	ERR_NOERR	Condition d'erreur Null
DBPLUS_ERR_DUPLICATE (entier)	ERR_DUPLICATE	Tentative d'insertion d'un doublon
DBPLUS_ERR_EOSCAN (entier)	ERR_EOSCAN	Fin de scan rget()
DBPLUS_ERR_EMPTY (entier)	ERR_EMPTY	Relation vide (serveur)
DBPLUS_ERR_CLOSE (entier)	ERR_CLOSE	Le serveur ne peut fermer
DBPLUS_ERR_WLOCKED (entier)	ERR_WLOCKED	La ligne est verrouillée en écriture
DBPLUS_ERR_LOCKED (entier)	ERR_LOCKED	La relation est déjà verrouillée
DBPLUS_ERR_NOLOCK (entier)	ERR_NOLOCK	La relation ne peut être verrouillée
DBPLUS_ERR_READ (entier)	ERR_READ	Erreur de lecture sur la relation
DBPLUS_ERR_WRITE (entier)	ERR_WRITE	Erreur d'écriture sur la relation
DBPLUS_ERR_CREATE (entier)	ERR_CREATE	La fonction système Create() a échoué
DBPLUS_ERR_LSEEK (entier)	ERR_LSEEK	La fonction système lseek() a échoué
DBPLUS_ERR_LENGTH (entier)	ERR_LENGTH	La ligne excède la taille maximale
DBPLUS_ERR_OPEN (entier)	ERR_OPEN	La fonction système Open() a échoué
DBPLUS_ERR_WOPEN (entier)	ERR_WOPEN	La relation est déjà ouverte en lecture
DBPLUS_ERR_MAGIC (entier)	ERR_MAGIC	Le fichier n'est pas une relation
DBPLUS_ERR_VERSION (entier)	ERR_VERSION	Le fichier est une très vieille relation
DBPLUS_ERR_PGFSIZE (entier)	ERR_PGFSIZE	La relation utilise une page de taille différente
DBPLUS_ERR_CRC (entier)	ERR_CRC	CRC invalide de la superpage
DBPLUS_ERR_PIPE (entier)	ERR_PIPE	Une relation pipée requiert lseek()
DBPLUS_ERR_NIDX (entier)	ERR_NIDX	Trop d'index secondaires
DBPLUS_ERR_MALLOC (entier)	ERR_MALLOC	L'appel à malloc() a échoué
DBPLUS_ERR_NUSERS (entier)	ERR_NUSERS	Erreur lors de l'utilisation du nombre maximal d'utilisateurs
DBPLUS_ERR_PREEXIT (entier)	ERR_PREEXIT	Causée par une utilisation invalide
DBPLUS_ERR_ONTRAP (entier)	ERR_ONTRAP	Causée par un signal
DBPLUS_ERR_PREPROC (entier)	ERR_PREPROC	Erreur dans le préprocesseur

DBPLUS_ERR_DBPARSE (<u>entier</u>)	ERR_DBPARSE	Erreur dans l'analyseur
DBPLUS_ERR_DBRUNERR (<u>entier</u>)	ERR_DBRUNERR	Erreur durant l'exécution
DBPLUS_ERR_DBPREEXIT (<u>entier</u>)	ERR_DBPREEXIT	La condition d'exit a été causée par la procédure prexit()
DBPLUS_ERR_WAIT (<u>entier</u>)	ERR_WAIT	Attend un peu (Simple seulement)
DBPLUS_ERR_CORRUPT_TUPLE (<u>entier</u>)	ERR_CORRUPT_TUPLE	Un client a envoyé une ligne corrompue
DBPLUS_ERR_WARNING0 (<u>entier</u>)	ERR_WARNING0	Les routines Simple ont rencontré une erreur non fatale, qui a été corrigée.
DBPLUS_ERR_PANIC (<u>entier</u>)	ERR_PANIC	Le serveur ne devrait pas s'arrêter, mais, après un désastre, envoyer ERR_PANIC à tout ses clients
DBPLUS_ERR_FIFO (<u>entier</u>)	ERR_FIFO	Impossible de créer une pile FIFO
DBPLUS_ERR_PERM (<u>entier</u>)	ERR_PERM	Permission refusée
DBPLUS_ERR_TCL (<u>entier</u>)	ERR_TCL	TCL_error
DBPLUS_ERR_RESTRICTED (<u>entier</u>)	ERR_RESTRICTED	Seulement deux utilisateurs
DBPLUS_ERR_USER (<u>entier</u>)	ERR_USER	Une erreur est survenue dans l'utilisation de la bibliothèque par l'application.
DBPLUS_ERR_UNKNOWN (<u>entier</u>)	ERR_UNKNOWN	Erreur inconnue

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [dbplus_add](#)
- [dbplus_aql](#)
- [dbplus_chdir](#)
- [dbplus_close](#)
- [dbplus_curr](#)
- [dbplus_errcode](#)
- [dbplus_erno](#)
- [dbplus_find](#)
- [dbplus_first](#)
- [dbplus_flush](#)
- [dbplus_freealllocks](#)
- [dbplus_freelock](#)
- [dbplus_freerlocks](#)
- [dbplus_getlock](#)
- [dbplus_getunique](#)
- [dbplus_info](#)
- [dbplus_last](#)
- [dbplus_lockrel](#)
- [dbplus_next](#)
- [dbplus_open](#)
- [dbplus_prev](#)
- [dbplus_rchperm](#)

- [dbplus_rcreate](#)
- [dbplus_rctxact](#)
- [dbplus_rctlke](#)
- [dbplus_resolve](#)
- [dbplus_restorepos](#)
- [dbplus_rkeys](#)
- [dbplus_ropen](#)
- [dbplus_rquery](#)
- [dbplus_rename](#)
- [dbplus_rsecindex](#)
- [dbplus_runlink](#)
- [dbplus_rzap](#)
- [dbplus_savepos](#)
- [dbplus_setindex](#)
- [dbplus_setindexbynumber](#)
- [dbplus_sql](#)
- [dbplus_tcl](#)
- [dbplus_tremove](#)
- [dbplus_undo](#)
- [dbplus_undoprepere](#)
- [dbplus_unlockrel](#)
- [dbplus_unselect](#)
- [dbplus_update](#)
- [dbplus_xlockrel](#)
- [dbplus_xunlockrel](#)

8.24.8 dbplus_aql() : Exécute une requête AQL

resource **dbplus_aql** (string query , string server , string dbpath)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[dbplus_aql](#) exécute la requête AQL query sur le serveur server avec le chemin dbpath .

En cas de succès, la fonction retournera une ressource de relation. Les données de résultat sont accessibles avec les fonctions [dbplus_next](#) et [dbplus_current](#) . Les autres fonctions d'accès aux relations ne fonctionneront pas avec une relation de résultat.

Plus d'informations sur le langage de requête AQL A... sont disponibles dans le manuel DB++.

8.24.9 dbplus_chdir() : Lit/modifie le chemin courant de base

string **dbplus_chdir** (string newdir)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_chdir` va changer le chemin de base courant utilisé par `dbplus_open` pour rechercher les fichiers de relations. `dbplus_chdir` retourne le chemin absolu courant. En appelant `dbplus_chdir` sans paramètre, vous pouvez lire le nom du dossier courant sans le modifier.

8.24.10 `dbplus_close()` : Ferme une relation

mixed `dbplus_close` (resource relation)

Attention
 Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

L'appel de `dbplus_close` fermera une relation précédemment ouverte par `dbplus_open` .

Retourne TRUE en cas de réussite ou DBPLUS_ERR_UNKNOWN en cas d'échec.

8.24.11 `dbplus_curr()` : Lit la ligne courante dans une relation

int `dbplus_curr` (resource relation , array tuple)

Attention
 Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_curr` lit les données de la ligne courante pour la relation `relation` et les retourne sous forme de tableau associatif, dans le paramètre `tuple` .

`dbplus_curr` va retourner zéro (alias DBPLUS_ERR_NOERR) en cas de succès, ou bien un code d'erreur DB++ en cas d'échec. Voyez la fonction `dbplus_errcode` ou l'introduction de ce chapitre pour plus d'informations sur les codes d'erreurs.

Voir aussi `dbplus_first` , `dbplus_prev` , `dbplus_next` et `dbplus_last` .

8.24.12 `dbplus_errcode()` : Obtenir un message d'erreur

string `dbplus_errcode` (int *errno*)

Attention
 Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_errcode` retourne un message d'erreur correspondant au code d'erreur `errno` , ou bien au dernier message d'erreur généré si aucun code d'erreur n'est fourni.

8.24.13 `dbplus_errno()` : Lit le code d'erreur généré par la dernière opération

int `dbplus_errno` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_erno` retourne le code d'erreur généré par la dernière opération DB++.

Voir aussi `dbplus_errcode` .

8.24.14 dbplus_find() : Ajoute une contrainte sur une relation

int `dbplus_find` (resource relation , array constraints , mixed tuple)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_find` ajoute la contrainte constraints sur la relation relation . Les appels ultérieurs à `dbplus_curr` ou `dbplus_next` ne retourneront que des lignes qui satisfont aux contraintes exprimées.

Les contraintes sont des triplets de chaînes contenant un nom de domaine, un opérateur de comparaison et une valeur de comparaison. Le paramètre constraints est un tableau qui est constitué d'une collection de tableaux de chaînes, chacun contenant un domaine, un opérateur et une valeur, ou bien un tableau contenant un multiple de trois éléments.

Les opérateurs de comparaisons sont les suivants : ' == ', ' > ', ' >= ', ' < ', ' <= ', ' != ', ' ~ ' pour les recherches classiques, et ' BAND ' ou ' BOR ' pour les opérations bits à bits.

Voir aussi `dbplus_unselect` .

8.24.15 dbplus_first() : Lit la première ligne d'une relation

int `dbplus_first` (resource relation , array tuple)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_curr` lit les données de la première ligne dans la relation relation , et les place dans le tableau associatif tuple .

`dbplus_curr` va retourner zéro (alias DBPLUS_ERR_NOERR) en cas de succès, ou bien un code d'erreur DB++ en cas d'échec. Voyez la fonction `dbplus_errcode` ou l'introduction de ce chapitre pour plus d'informations sur les codes d'erreurs.

Voir aussi `dbplus_first` , `dbplus_prev` , `dbplus_next` et `dbplus_last` .

8.24.16 dbplus_flush() : Ecrit toutes les modifications apportées à une relation

int **dbplus_flush** (resource relation)

Attention

<p>Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.</p>

dbplus_flush va écrire sur le disque toutes les modifications apportées à relation depuis la dernière écriture.

dbplus_flush va retourner zéro (alias DBPLUS_ERR_NOERR) en cas de succès, ou bien un code d'erreur DB++ en cas d'échec. Voyez la fonction dbplus_errcode ou l'introduction de ce chapitre pour plus d'informations sur les codes d'erreurs.

8.24.17 dbplus_freealllocks() : Libère tous les verrous posés par le client

int **dbplus_freealllocks** (void)

Attention

<p>Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.</p>

dbplus_freealllocks libère tous les verrous posés par le client.

Voir aussi dbplus_getlock , dbplus_freelock et dbplus_freerlocks .

8.24.18 dbplus_freelock() : Lève un verrou en écriture sur une ligne

int **dbplus_freelock** (resource relation , string tname)

Attention

<p>Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.</p>

dbplus_freelock supprime le verrou en écriture sur la ligne tuple , posé au préalable par la fonction dbplus_getlock .

Voir aussi dbplus_getlock , dbplus_freerlocks et dbplus_freealllocks .

8.24.19 dbplus_freerlocks() : Lève tous les verrous sur les lignes d'une relation

int **dbplus_freerlocks** (resource relation)

Attention

<p>Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.</p>

`dbplus_freelocks` lève tous les verrous sur les lignes de la relation relation .

Voir aussi `dbplus_getlock` , `dbplus_freelock` et `dbplus_freealllocks` .

8.24.20 `dbplus_getlock()` : Pose un verrou sur une ligne

int `dbplus_getlock` (resource relation , string tname)

Attention
 Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_getlock` tente de poser un verrou en écriture sur la ligne tuple . Cette fonction retournera zéro en cas de succès, ou bien un code d'erreur (spécifiquement `DBPLUS_ERR_WLOCKED`) en cas d'erreur.

Voir aussi `dbplus_freelock` , `dbplus_freelocks` et `dbplus_freealllocks` .

8.24.21 `dbplus_getunique()` : Retourne un identifiant unique pour une relation

int `dbplus_getunique` (resource relation , int uniqueid)

Attention
 Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_getunique` demande un nombre, qui sera garanti comme unique pour la relation relation , et le retourne dans le paramètre uniqueid .

`dbplus_getunique` va retourner zéro (alias `DBPLUS_ERR_NOERR`) en cas de succès, ou bien un code d'erreur `DB++` en cas d'échec. Voyez la fonction `dbplus_errcode` ou l'introduction de ce chapitre pour plus d'informations sur les codes d'erreurs.

8.24.22 `dbplus_info()` : Récupère les informations sur une relation

int `dbplus_info` (resource relation , string key , array result)

Attention
 Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention
 Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.24.23 `dbplus_last()` : Lit la dernière ligne d'une relation

int `dbplus_last` (resource relation , array tuple)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_curr` lit les données de la dernière ligne de la relation `relation` , et en fait la ligne courante avant de placer les informations dans le tableau tuple .

`dbplusdbplus_curr_add` va retourner zéro (alias `DBPLUS_ERR_NOERR`) en cas de succès, ou bien un code d'erreur DB++ en cas d'échec. Voyez la fonction `dbplus_errcode` ou l'introduction de ce chapitre pour plus d'informations sur les codes d'erreurs.

Voir aussi `dbplus_first` , `dbplus_curr` , `dbplus_prev` et `dbplus_next` .

8.24.24 dbplus_lockrel() : Pose un verrou en écriture sur une relation

int `dbplus_lockrel` (resource relation)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_lockrel` va tenter de poser un verrou en écriture sur la relation `relation` . Les autres clients pourront toujours lire dans cette relation, mais devront attendre pour la modifier.

8.24.25 dbplus_next() : Lit la ligne suivante dans la relation

int `dbplus_next` (resource relation , array tuple)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_curr` lit les données de la ligne suivante dans la relation , et en fait la ligne courante avant de placer les données dans le paramètre tuple .

`dbplus_curr` va retourner zéro (alias `DBPLUS_ERR_NOERR`) en cas de succès, ou bien un code d'erreur DB++ en cas d'échec. Voyez la fonction `dbplus_errcode` ou l'introduction de ce chapitre pour plus d'informations sur les codes d'erreurs.

Voir aussi `dbplus_first` , `dbplus_curr` , `dbplus_prev` et `dbplus_last` .

8.24.26 dbplus_open() : Ouvre un fichier de relation

resource `dbplus_open` (string name)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Le fichier de relation name sera ouvert. name peut être un nom de fichier, ou un chemin relatif ou absolu. Dans tous les cas, il sera remplacé par un chemin absolu, sur une machine spécifique et sur un serveur spécifique.

En cas de succès, une ressource de fichier de relation (dite aussi curseur) sera retournée. Elle sera alors utilisée avec les fonctions utilisant cette relation. En cas d'échec, la valeur zéro est retournée et le code d'erreur peut être demandé via la fonction [dbplus_errno](#) .

8.24.27 dbplus_prev() : Lit la ligne précédente dans la relation

int **dbplus_prev** (resource relation , array tuple)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[dbplus_curr](#) lit les données dans la ligne précédant la ligne courante dans la relation relation ; en fait la ligne courante et passe les données représentant la ligne sous la forme d'un tableau associatif dans le paramètre tuple .

Cette fonction retourne zéro (e.g. DBPLUS_ERR_NOERR) en cas de succès ou un code d'erreur db++ en cas d'échec. Voyez [dbplus_errcode](#) ou l'introduction de ce chapitre pour plus de détails sur les codes d'erreur db++.

Voir aussi [dbplus_first](#) , [dbplus_curr](#) , [dbplus_next](#) et [dbplus_last](#) .

8.24.28 dbplus_rchperm() : Modifie les droits sur la relation

int **dbplus_rchperm** (resource relation , int mask , string user , string group)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[dbplus_rchperm](#) modifie les droits d'accès à la relation relation . Les nouveaux droits sont spécifiés avec les paramètres mask , user et group . Les valeurs à fournir sont spécifiques à chaque système d'exploitation.

8.24.29 dbplus_rcreate() : Crée une nouvelle relation DB++

resource **dbplus_rcreate** (string name , mixed domlist , bool overwrite)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[dbplus_rcreate](#) crée une nouvelle relation nommée name . Une autre relation du même nom sera écrasée si la relation n'est pas utilisée à ce moment, et si overwrite vaut TRUE .

domlist doit contenir les spécifications de domaine pour la nouvelle relation, sous la forme d'un tableau de chaînes descriptives. (`dbplus_rcreate` accepte aussi une chaîne avec des domaines séparés par des espaces, mais l'utilisation de cette technique est découragée). Une description de domaine est constituée d'un nom de domaine unique pour cette relation, d'un slash "/" et d'un caractère de spécification de type. Reportez-vous à la documentation DB++, et en particulier à la page concernant `dbcreate(1)`, pour connaître le détail des spécificateurs de type et leur signification.

8.24.30 `dbplus_rctexact()` : Crée une copie exacte mais vide d'une relation

mixed `dbplus_rctexact` (string name , resource relation , bool overwrite)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_rctexact` crée une copie exacte de la relation relation , sous le nouveau nom de name . Une relation qui porterait déjà le nom de name ne sera écrasée que si `overwrite` vaut TRUE et qu'aucun autre processus ne l'utilise.

Retourne TRUE en cas de réussite ou DBPLUS_ERR_UNKNOWN en cas d'échec.

8.24.31 `dbplus_rctllike()` : Crée une copie vide d'une relation avec les index

mixed `dbplus_rctllike` (string name , resource relation , int overwrite)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_rctllike` crée une copie vide de la relation relation , sous le nouveau nom de name , mais avec les index par défaut. Une relation qui porterait déjà le nom de name ne sera écrasée que si `overwrite` vaut TRUE et qu'aucun autre processus ne l'utilise.

Retourne TRUE en cas de réussite ou DBPLUS_ERR_UNKNOWN en cas d'échec.

8.24.32 `dbplus_resolve()` : Résoud les informations d'hôte pour une relation

array `dbplus_resolve` (string relation_name)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_resolve` résoud le nom de relation relation_name et collecte le nom interne de serveur, le nom d'hôte et le chemin vers la base sur cet hôte. La fonction retourne un tableau contenant ces valeurs, sous les index ' sid ', ' host ' et ' host_path ', ou FALSE en cas d'erreur.

Voir aussi [dbplus_tcl](#) .

8.24.33 dbplus_restorepos() : Restaure la position

int **dbplus_restorepos** (resource relation , array tuple)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.24.34 dbplus_rkeys() : Spécifie la nouvelle clé primaire d'une relation

mixed **dbplus_rkeys** (resource relation , mixed domlist)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
<u>dbplus_rkeys</u> remplace la clé primaire courante de la relation relation avec la combinaison de domaines spécifiée par domlist .

domlist peut être passé comme un nom de domaine seul, ou comme un tableau de noms de domaines.

Retourne TRUE en cas de réussite ou DBPLUS_ERR_UNKNOWN en cas d'échec.

8.24.35 dbplus_ropen() : Ouvre un fichier de relation local

resource **dbplus_ropen** (string name)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
<u>dbplus_ropen</u> ouvre le fichier de relation file localement, pour accès rapide, sans les coûts supplémentaires du système client/serveur. L'accès est alors en lecture seule, et seules <u>dbplus_current</u> et <u>dbplus_next</u> peuvent être utilisées avec la relation retournée.

8.24.36 dbplus_rquery() : Accomplit une requête AQL locale

resource **dbplus_rquery** (string query , string dbpath)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS

PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_rquery` accomplit une requête AQL brute, en utilisant l'interpréteur de la bibliothèque du client DB++. `dbplus_rquery` est plus rapide que `dbplus_aql` mais ne fonctionne que sur des données locales.

8.24.37 `dbplus_rrename()` : Renomme une relation

int `dbplus_rrename` (resource relation , string name)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_rrename` va changer le nom de la relation en name .

8.24.38 `dbplus_rsecindex()` : Crée un nouvel index secondaire pour une relation

mixed `dbplus_rsecindex` (resource relation , mixed domlist , int type)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_rsecindex` crée un nouvel index secondaire pour la relation relation , qui sera constitué par domlist et sera de type type

domlist peut être passé comme un nom de domaine seul, ou comme un tableau de noms de domaines.

Retourne TRUE en cas de réussite ou DBPLUS_ERR_UNKNOWN en cas d'échec.

8.24.39 `dbplus_runlink()` : Détruit une relation

int `dbplus_runlink` (resource relation)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_unlink` ferme et supprime la relation relation .

8.24.40 `dbplus_rzap()` : Supprime toutes les lignes d'une relation

int `dbplus_rzap` (resource relation)

Attention

<p>Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.</p>

`dbplus_rzap` supprime toutes les lignes de la relation relation .

8.24.41 `dbplus_savepos()` : Sauvegarde la position

int `dbplus_savepos` (resource relation)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.24.42 `dbplus_setindex()` : Définie l'index

int `dbplus_setindex` (resource relation , string idx_name)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.24.43 `dbplus_setindexbynumber()` : Définie l'index par un nombre

int `dbplus_setindexbynumber` (resource relation , int idx_number)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.24.44 `dbplus_sql()` : Exécute une requête SQL

resource `dbplus_sql` (string query , string server , string dbpath)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.24.45 `dbplus_tcl()` : Exécute du code TCL côté serveur

string `dbplus_tcl` (int `sid` , string `script`)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Un serveur db++ va préparer un interpréteur TCL pour chaque connexion de client. Cet interpréteur va permettre au serveur d'exécuter du code TCL fourni par le client, un peu comme une procédure stockée, pour améliorer les performances du serveur en évitant des transferts client/serveur et des changements de contexte.

`dbplus_tcl` doit recevoir l'identifiant de connexion `sid` et le code TCL `script` qui sera exécuté. `dbplus_resolve` fournira l'identifiant de connexion. `dbplus_tcl` retourne ce que le code TCL retourne, ou bien un message d'erreur TCL si le code TCL échoue.

Voir aussi `dbplus_resolve` .

8.24.46 `dbplus_tremove()` : Supprime la ligne courante et retourne la nouvelle ligne courante

int `dbplus_tremove` (resource `relation` , array `tuple` , array `current`)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`dbplus_tremove` supprime la ligne `tuple` de la relation si cela correspond parfaitement à une ligne dans la relation. `current` , si fourni, contient les données de la nouvelle ligne, une fois la suppression effectuée avec `dbplus_tremove` .

8.24.47 `dbplus_undo()` : Annule

int `dbplus_undo` (resource `relation`)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.24.48 `dbplus_undoprepere()` : Prépare l'annulation

int `dbplus_undoprepere` (resource `relation`)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.24.49 **dbplus_unlockrel()** : Libère un verrou en écriture posé sur une relation

int **dbplus_unlockrel** (resource relation)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

dbplus_unlockrel lève le verrou en écriture, posé par la fonction dbplus_lockrel sur la relation relation .

8.24.50 **dbplus_unselect()** : Retire une contrainte d'une relation

int **dbplus_unselect** (resource relation)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

dbplus_unselect supprime la contrainte posée par dbplus_find sur la relation .

8.24.51 **dbplus_update()** : Modifie une ligne dans une relation

int **dbplus_update** (resource relation , array old , array new)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

dbplus_update remplace la ligne identifiée par old avec les données du paramètre new si et seulement si old identifie exactement une ligne dans une relation .

8.24.52 **dbplus_xlockrel()** : Pose un verrou exclusif sur une relation

int **dbplus_xlockrel** (resource relation)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

dbplus_xlockrel pose un verrou exclusif sur la relation , ce qui empêchera les autres clients d'accéder à la relation, et même de la lire.

Voir aussi dbplus_xunlockrel .

8.24.53 **dbplus_xunlockrel()** : Lève un verrou exclusif posé sur une relation

int **dbplus_xunlockrel** (resource relation)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

dbplus_xunlockrel lève le verrou exclusif, posé sur la relation , grâce à la fonction dbplus_xlockrel .

8.25 Fonctions dbx

8.25.1 Introduction

Le module dbx est un module d'abstraction de base de données (db pour database (base de données) et 'X' pour toutes les bases supportées). Les fonctions dbx vous permettent d'accéder à toutes les bases supportées, avec la même convention. Les fonctions dbx elles-mêmes ne s'interfaçent pas directement avec les bases de données, mais s'interfaçent avec les modules utilisées pour supporter ces bases.

Note

Cette extension a été déplacée dans le module PECL et ne sera plus intégrée dans PHP à partir de PHP 5.1.0.

8.25.2 Pré-requis

Pour pouvoir utiliser une base de données avec le module dbx, le module doit être soit lié, soit chargé dans PHP et le module de base de données doit être supporté par le module dbx. Actuellement les bases suivantes sont supportées, et d'autres suivront :

- FrontBase (disponible depuis PHP 4.1.0).
- Microsoft SQL Server
- MySQL
- ODBC
- PostgreSQL
- Sybase-CT (disponible depuis PHP 4.2.0).
- Oracle (oci8) (disponible depuis PHP 4.3.0).
- SQLite (PHP 5).

La documentation nécessaire pour ajouter le support d'autres bases de données est disponible à l'adresse <http://www.guidance.nl/php/dbx/doc/> .

8.25.3 Installation

Il vous faut avoir ces fonctions compilées avec PHP (option de configuration --enable-dbx et toutes les bases que vous souhaitez utiliser. Par exemple, si vous voulez accéder à MySQL depuis dbx, vous devez aussi configurer PHP avec --with-mysql=[DIR] .

8.25.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

Nom	Par défaut	Modifiable	Historique
dbx.colnames_case	"unchanged"	PHP_INI_SYSTEM	Disponible depuis PHP 4.3.0.

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à Directives du php.ini .

Voici un éclaircissement sur l'utilisation des directives de configuration.

```
dbx.colnames_case string
```

Les noms de colonnes peuvent être retournés comme "inchangés" ("unchanged"), ou bien "majuscules" ("uppercase") ou encore "minuscules" ("lowercase"). Cette directive peut être reconfigurée par une option de dbx_query .

8.25.5 Types de ressources

Il y a deux ressources utilisées par le module dbx. Le premier est le type link-object , pour les connexions aux bases de données, et l'autre est le type result-object qui contient les lignes lues.

8.25.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

DBX_MYSQL (entier)
DBX_ODBC (entier)
DBX_PGSQL (entier)
DBX_MSSQL (entier)
DBX_FBSQL (entier)
DBX_OCI8 (entier) (disponible à partir de PHP 4.3.0)
DBX_SYBASECT (entier)
DBX_SQLITE (entier) (PHP 5)
DBX_PERSISTENT (entier)
DBX_RESULT_INFO (entier)
DBX_RESULT_INDEX (entier)
DBX_RESULT_ASSOC (entier)
DBX_RESULT_UNBUFFERED (entier) (PHP 5)
DBX_COLNAMES_UNCHANGED (entier) (uniquement disponible à partir de PHP 4.3.0)
DBX_COLNAMES_UPPERCASE (entier) (uniquement disponible à partir de PHP 4.3.0)
DBX_COLNAMES_LOWERCASE (entier) (uniquement disponible à partir de PHP 4.3.0)
DBX_CMP_NATIVE (entier)
DBX_CMP_TEXT (entier)
DBX_CMP_NUMBER (entier)
DBX_CMP_ASC (entier)
DBX_CMP_DESC (entier)

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [dbx_close](#)
- [dbx_compare](#)
- [dbx_connect](#)
- [dbx_error](#)
- [dbx_escape_string](#)
- [dbx_fetch_row](#)
- [dbx_query](#)
- [dbx_sort](#)

8.25.8 dbx_compare() : Compare deux lignes DBX afin de les trier

int **dbx_compare** (array row_a , array row_b , string column_key , int flags)

dbx_compare retourne 0 si row_a[\$column_key] est égale à row_b[\$column_key] , et 1 ou -1 si le premier est plus grand ou plus petit que le dernier, respectivement, ou bien le contraire si l'option flags est configurée à DBX_CMP_DESC . dbx_compare est une fonction d'aide pour dbx_sort , afin d'aider aux tris.

Le paramètre flags peut prendre plusieurs valeurs :

- DBX_CMP_ASC - ordre ascendant
- DBX_CMP_DESC - ordre descendant

et le type utilisé pour la comparaison :

- DBX_CMP_NATIVE - par de conversion de type
- DBX_CMP_TEXT - compare les lignes comme des chaînes
- DBX_CMP_NUMBER - compare les lignes comme des nombres

Une direction de tri et un type de comparaison peuvent être combinés avec l'opérateur OR (|). La valeur par défaut pour l'option flags est DBX_CMP_ASC | DBX_CMP_NATIVE .

Exemple avec dbx_compare

```
<?php
function user_re_order ($a, $b) {
    $rv = dbx_compare ($a, $b, "parentid", DBX_CMP_DESC);
    if ( !$rv ) {
        $rv = dbx_compare ($a, $b, "id", DBX_CMP_NUMBER);
    }
    return $rv;
}

$link = dbx_connect (DBX_ODBC, "", "db", "username", "password")
or die ("Impossible de se connecter");

$result = dbx_query ($link, "SELECT id, parentid, description FROM table ORDER BY id");
// les données de $result sont désormais classées par id

dbx_sort ($result, "user_re_order");
// la date dans $result est maintenant ordonnée par parentid (descending), puis par id

dbx_close ($link);
?>
```

Voir aussi dbx_sort .

8.25.9 dbx_connect() : Ouvre une connexion à une base de données

object **dbx_connect** (mixed module , string host , string database , string username , string password , int persistent)

dbx_connect retourne une ressource dbx_link_object en cas de succès, FALSE sinon. Si la connexion a put être établie, mais que la base de données n'a pas pu être sélectionnée, la fonction retournera quand même une ressource. Le paramètre persistent peut prendre la valeur DBX_PERSISTENT , pour créer une connexion persistente.

Le paramètre module peut être soit une chaîne, soit une constante. Les valeurs possibles de module sont listées ci-dessous (n'oubliez pas que cela fonctionnera que si le module associé est chargé) :

- module DBX_MYSQL : "mysql"
- module DBX_ODBC : "odbc"
- module DBX_PGSQL : "pgsql"
- module DBX_MSSQL : "mssql"
- module DBX_FBSQL : "fbsql" (disponible depuis PHP 4.1.0)
- module DBX_SYBASECT : "sybase_ct" (disponible depuis PHP 4.2.0)
- module DBX_OCI8 : "oci8" (disponible depuis PHP 4.3.0)
- module DBX_SQLITE : "sqlite" (PHP 5)

Les paramètres host , database , username et password sont attendus, mais ne sont pas toujours utiles, suivant la fonction de connexion de la base de données utilisée.

L'objet retourné object a trois propriétés :

database

Nom de la base de données sélectionnées

handle

Ressource de connexion de la base de données, et donc, elle peut être utilisée dans les fonctions spécifiques au module (si requis).

Utilisation de la ressource de base de données DBX

```
<?php
$link = dbx_connect(DBX_MYSQL, "localhost", "db", "username", "password");
mysql_close($link->handle);
// dbx_close($link) serait plus appropriée ici.
?>
```

module

Utiliser en interne par dbx uniquement et actuellement, correspond au numéro du module mentionné ci-dessus.

Exemple avec dbx_connect

```
<?php
$link = dbx_connect(DBX_ODBC, "", "db", "username", "password", DBX_PERSISTENT)
    or die("Impossible de se connecter");

echo "Connexion réussie";
dbx_close($link);
?>
```

Note

Reportez-vous à la documentation du module utilisé.

Voir aussi dbx_close .

8.25.10 dbx_error() : Rapporte le message d'erreur du dernier appel de fonction DBX

string **dbx_error** (object link_identifiant)

dbx_error retourne une chaîne contenant le message d'erreur du module sélectionné. S'il y a des connexions multiples sur le même module, seule la dernière erreur est fournie. S'il a des

connexions sur différents modules, la dernière erreur du module est retournée (le module est alors représenté par `link_identifier`).

Exemple avec `dbx_error`

```
<?php
$link = dbx_connect("mysql", "localhost", "base de données", "utilisateur", "mot de passe")
      or die ("Impossible de se connecter");
$result = dbx_query($link, "select id from nonexistingtbl");
if ($result==0) {
    echo dbx_error($link);
}
dbx_close($link);
?>
```

Note

Reportez-vous aussi à la documentation de la base de données que vous utilisez.

Le message de Microsoft SQL Server est en fait le résultat de la fonction `mssql_get_last_message`.

Le support des messages d'erreurs Oracle (`oci8`) ne sont pas implémentés (pour le moment).

8.25.11 `dbx_escape_string()` : Protège une chaîne de caractères pour l'utiliser dans une requête

string `dbx_escape_string` (object `link_identifier` , string `text`)

`dbx_escape_string` retourne le texte `text` dont les caractères spéciaux ont été protégés (les guillemets, les anti-slash...). En cas d'erreur, `dbx_escape_string` retourne NULL.

Exemple avec `dbx_escape_string`

```
<?php
$link = dbx_connect(DBX_MYSQL, "localhost", "db", "username", "password")
      or die ("Impossible de se connecter");

$text = dbx_escape_string($link, "C'est une chaîne avec des anti slash (\\).");
$result = dbx_query($link, "insert into tbl (txt) values ('.$text.')");
if ( $result == 0 ) {
    echo dbx_error ($link);
}
dbx_close ($link);
?>
```

Voir aussi `dbx_query`.

8.25.12 `dbx_fetch_row()` : Lit une ligne dans un résultat DBX ayant l'option `DBX_RESULT_UNBUFFERED` activée

mixed `dbx_fetch_row` (object `result_identifier`)

`dbx_fetch_row` retourne une ligne en cas de succès, et un 0 en cas d'échec (e.g. lorsqu'aucune ligne n'est disponible). Lorsque l'option `DBX_RESULT_UNBUFFERED` n'est pas activée dans la requête, `dbx_fetch_row` va échouer car toutes les lignes auront été déjà lues dans le membre `data`.

De plus, le membre `rows` de l'objet de résultat est incrémenté à chaque lecture réussie de `dbx_fetch_row`.

Comment gérer les valeurs retournées par `dbx_fetch_row`

```
<?php
$result = dbx_query ($link, 'SELECT id, parentid, description FROM table', DBX_RESULT_UNBUFFERED);

echo "<table>\n";
while ( $row = dbx_fetch_row($result) ) {
    echo "<tr>\n";
    foreach ( $row as $field ) {
        echo "<td>$field</td>";
    }
    echo "</tr>\n";
}
echo "</table>\n";
?>
```

Le paramètre `result_identifier` est le résultat de la fonction `dbx_query`.

L'objet retourné contient les mêmes informations que n'importe quelle ligne d'un résultat dbx, placé dans le membre `data`, incluant l'accès aux colonnes par index ou par nom, suivant l'utilisation des options avec `dbx_query`.

Voir aussi `dbx_query`.

8.25.13 `dbx_query()` : Envoie une requête et lit tous les résultats DBX

mixed `dbx_query` (object `link_identifier` , string `sql_statement` , int `flags`)

`dbx_query` retourne un objet `dbx_result_object` ou 1 en cas de succès (un objet de résultat ne sera retourné que pour les requêtes SQL qui retournent un résultat), ou 0 en cas d'erreur. L'objet résultant n'est retourné que si la requête donnée par `sql_statement` produit un jeu d'enregistrements. (i.e. une requête SELECT, même si le résultat est vide). La requête est exécuté sur la connexion `link_identifier`.

Comment gérer la valeur retournée par `dbx_query`

```
<?php
$link = dbx_connect(DBX_ODBC, "", "db", "username", "password")
      or die("Impossible de se connecter");

$result = dbx_query($link, 'SELECT id, parentid, description FROM table');

if ( is_object($result) ) {
    // ... Faire quelque chose d'utile ici
    // premièrement, afficher les noms des champs et leurs types
    // et tracer un tableau avec les valeurs retournées
} else {
    exit("La requête a échoué");
}

dbx_close($link);
?>
```

Le paramètre `flags` sert à contrôler la quantité d'informations retournée. Il peut être n'importe quelle combinaisons par l'opérateur OR des constantes ci-après. Les constantes remplacent la configuration du `php.ini`.

8.25.12 `dbx_fetch_row()` : Lit une ligne dans un résultat DBX ayant l'option `DBX_RESULT_UNBUFFERED`

DBX_RESULT_INDEX

Ce paramètre est **toujours** activé, c'est à dire que l'objet retourné dispose du membre data qui est un tableau à deux dimensions, indexé numériquement. Par exemple, dans l'expression data[2][3] 2 représente le numéro de ligne et 3 représente le numéro de colonne. Les premières lignes et colonnes sont indexées à 0.

Si DBX_RESULT_ASSOC est aussi spécifié, l'objet retourné contient en plus les informations liées à DBX_RESULT_INFO, même s'il n'a pas été spécifié.

DBX_RESULT_INFO

Cette option fournit des informations sur les colonnes, comme les noms des champs et leur type.

DBX_RESULT_ASSOC

Cette option fait que la valeur des champs peut être accessible avec le nom de la colonne, utilisé comme clé dans le tableau du membre data.

Les résultats ainsi associés sont des références sur les valeurs identifiées par les index numériques, ce qui fait que modifier data[0][0] fait que data[0]['nom_de_la_premiere_colonne'] sera aussi modifié.

DBX_RESULT_UNBUFFERED (PHP 5)

Cet option ne crée pas de propriété data et la propriété rows sera initialement positionnée à 0. Utilisez ce flag pour des jeux d'enregistrements importants et utilisez dbx_fetch_row pour retrouver le résultat ligne par ligne.

dbx_fetch_row retournera les lignes qui se conforment au flag de la requête. Par ailleurs, cela mettra à jour les lignes à chaque appel de la fonction.

DBX_COLNAMES_UNCHANGED (disponible depuis PHP 4.3.0)

La casse du nom des colonnes retournées ne sera pas modifiée.

DBX_COLNAMES_UPPERCASE (disponible depuis PHP 4.3.0)

Les noms de colonnes seront mis en majuscules.

DBX_COLNAMES_LOWERCASE (disponible depuis PHP 4.3.0)

Les noms de colonnes seront mis en minuscules.

Notez que DBX_RESULT_INDEX est toujours active, indépendamment de la valeur de flags. Cela signifie que seules les combinaisons suivantes sont utiles :

- DBX_RESULT_INDEX
- DBX_RESULT_INDEX | DBX_RESULT_INFO
- DBX_RESULT_INDEX | DBX_RESULT_INFO | DBX_RESULT_ASSOC - c'est la valeur par défaut, si flags est omis.

Un objet dbx_result_object a 5 membres (éventuellement 4, suivant les valeurs de flags) :

handle

Ceci est une ressource représentant la connexion à la base de données, et il peut être utilisé (si nécessaire) avec les fonctions spécialisées de la base.

Exemple avec dbx_query

```
<?php
$result = dbx_query($link, "SELECT id FROM table");
mysql_field_len($result->handle, 0);
?>
```

cols et rows

Ces deux membres contiennent respectivement le nombre de colonnes et de lignes.

Nombre de lignes et colonnes avec dbx_query

```
<?php
$result = dbx_query ($link, 'SELECT id FROM table');
echo $result->rows; // nombre de lignes
echo $result->cols; // nombre de champs
?>
```

info (optionnel)

Ce membre est retourné uniquement si DBX_RESULT_INFO ou DBX_RESULT_ASSOC sont spécifiés dans le paramètre flags . C'est un tableau à deux dimensions, avec deux lignes (name et type) pour lire les informations de colonnes.

Listes les types et noms de colonnes avec dbplus

```
<?php
$result = dbx_query ($link, 'SELECT id FROM table',
                    DBX_RESULT_INDEX | DBX_RESULT_INFO);

for ($i = 0; $i < $result->cols; $i++ ) {
    echo $result->info['name'][$i] . "\n";
    echo $result->info['type'][$i] . "\n";
}
?>
```

data

Ce membre contient les résultats obtenus, possiblement associés avec le nom de colonne, en fonction de la valeur du paramètre flags . Si DBX_RESULT_ASSOC est activé, il est possible d'utiliser aussi \$result->data[2]["nom de la colonne"] .

Afficher le contenu d'une base avec dbplus

```
<?php
$result = dbx_query ($link, 'SELECT id, parentid, description FROM table');

echo "<table>\n";
foreach ( $result->data as $row ) {
    echo "<tr>\n";
    foreach ( $row as $field ) {
        echo "<td>$field</td>";
    }
    echo "</tr>\n";
}
echo "</table>\n";
?>
```

Comment utiliser les requêtes UNBUFFERED

```
<?php

$result = dbx_query ($link, 'SELECT id, parentid, description FROM table', DBX_RESULT_

echo "<table>\n";
while ($row = dbx_fetch_row($result)) {
    echo "<tr>\n";
    foreach ($row as $field) {
        echo "<td>$field</td>";
    }
    echo "</tr>\n";
}
echo "</table>\n";

?>
```

Note

Reportez-vous aussi à la documentation de la base de données que vous utilisez.

Les noms des colonnes pour les requêtes sur des bases de données Oracle sont retournés en minuscules.

Voir aussi [dbx_fetch_row](#) , [dbx_escape_string](#) et [dbx_connect](#) .

8.25.14 [dbx_sort\(\)](#) : Trie un résultat avec une fonction utilisateur

bool **dbx_sort** (object result , string user_compare_function)

[dbx_sort](#) trie les lignes du résultat DBX result , à l'aide de la fonction PHP user_compare_function . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Il est toujours plus intéressant d'utiliser la clause SQL ORDER BY que la fonction [dbx_sort](#) , si possible.

Exemple avec [dbx_sort](#)

```
<?php
function user_re_order($a, $b)
{
    $rv = dbx_compare($a, $b, "parentid", DBX_CMP_DESC);
    if (!$rv) {
        $rv = dbx_compare($a, $b, "id", DBX_CMP_NUMBER);
    }
    return $rv;
}

$link = dbx_connect("odbc", "", "base de données", "utilisateur", "mot de passe")
    or die ("Impossible de se connecter");

$result = dbx_query($link, "SELECT id, parentid, description FROM tbl ORDER BY id");
    // Les données sont maintenant triées par id

dbx_sort($result, "user_re_order");
    // Les données sont maintenant triées par parentid décroissant, puis par id

dbx_close($link);
?>
```

Voir aussi [dbx_compare](#) .

8.26 Direct IO

8.26.1 Introduction

PHP supporte les entrées/sorties directes, comme décrites dans le Standard Posix (Section 6). Cela permet de réaliser des accès en lecture/écriture à un niveau inférieur aux fonctions proposées par le langage C (fopen, fread...). L'utilisation des fonctions DIO ne doit être envisagée que si le contrôle directe d'un périphérique est requis. Dans les autres cas, les fonctions système de fichier sont plus qu'adéquates.

Note
Cette extension a été déplacée dans le module <u>PECL</u> et ne sera plus intégrée dans PHP à partir de PHP 5.1.0.
Cette extension est disponible sur les plates-formes Windows depuis PHP 5.0.0.

8.26.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.26.3 Installation

Pour faire fonctionner ces fonctions, vous devez configurer PHP avec l'option `--enable-dio` .

8.26.4 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

- c (entier)
- F_DUPFD (entier)
- F_GETFD (entier)
- F_GETFL (entier)
- F_GETLK (entier)
- F_GETOWN (entier)
- F_RDLCK (entier)
- F_SETFL (entier)
- F_SETLK (entier)
- F_SETLKW (entier)
- F_SETOWN (entier)
- F_UNLCK (entier)
- F_WRLCK (entier)
- O_APPEND (entier)
- O_ASYNC (entier)
- O_CREAT (entier)
- O_EXCL (entier)
- O_NDELAY (entier)
- O_NOCTTY (entier)
- O_NONBLOCK (entier)

O_RDONLY ([entier](#))
O_RDWR ([entier](#))
O_SYNC ([entier](#))
O_TRUNC ([entier](#))
O_WRONLY ([entier](#))
S_IRGRP ([entier](#))
S_IROTH ([entier](#))
S_IRUSR ([entier](#))
S_IRWXG ([entier](#))
S_IRWXO ([entier](#))
S_IRWXU ([entier](#))
S_IWGRP ([entier](#))
S_IWOTH ([entier](#))
S_IWUSR ([entier](#))
S_IXGRP ([entier](#))
S_IXOTH ([entier](#))
S_IXUSR ([entier](#))

8.26.5 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.26.6 Types de ressources

Cette extension définit un type de ressource : un descripteur de fichier retourné par [dio_open](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Constantes pré-définies](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [dio_close](#)
- [dio_fcntl](#)
- [dio_open](#)
- [dio_read](#)
- [dio_seek](#)
- [dio_stat](#)
- [dio_tcsetattr](#)
- [dio_truncate](#)
- [dio_write](#)

8.26.8 dio_fcntl() : Exécute une fonction fichier de langage C

mixed **dio_fcntl** (resource *fd* , int *cmd* , *mixed args*)

[dio_fcntl](#) exécute l'opération *cmd* sur le fichier représenté par la ressource *fd* . Certaines commandes demandent des arguments supplémentaires qui sont fournis dans l'argument *args* .

Note

Cette fonction n'est pas implémentée sous Windows.
--

8.26.9 `dio_open()` : Ouvre un nouveau fichier avec les permissions spécifiés

resource `dio_open` (string filename , int flags , int mode)

`dio_open` ouvre un accès à un fichier et retourne une ressource de fichier.

8.26.10 `dio_read()` : Lit les octets dans un fichier

string `dio_read` (resource fd , int len)

`dio_read` lit et retourne len octets dans le fichier représenté par la ressource fd .

8.26.11 `dio_seek()` : Déplace le pointeur interne de fichier

int `dio_seek` (resource fd , int pos , int whence)

`dio_seek` sert à modifier la position du pointeur de fichier dans le fichier.

8.26.12 `dio_stat()` : Lit des informations sur le fichier

array `dio_stat` (resource fd)

`dio_stat` retourne les informations sur le fichier.

8.26.13 `dio_tcsetattr()` : Modifie les attributs du terminal et le taux de baud du port série

bool `dio_tcsetattr` (resource fd , array options)

`dio_tcsetattr` modifie les attributs du terminal et le taux de baud du port série de ressource .

Note

Cette fonction n'est pas implémentée sous Windows.
--

8.26.14 `dio_truncate()` : Tronque un fichier

bool `dio_truncate` (resource fd , int offset)

Si auparavant le fichier était plus grand que cette taille, les données en plus seront perdues. Si par contre le fichier était plus petit, il n'est pas spécifié si le fichier est gardé intact ou agrandi. Dans le dernier cas, la partie étendue devient des octets zéros.

Note

Cette fonction n'est pas implémentée sous Windows.
--

8.26.15 dio_write() : Écrit des données dans le fichier avec la possibilité de tronquer sa longueur

int **dio_write** (resource *fd* , string *data* , int *len*)

dio_write écrit jusqu'à *len* octets issus de la variable *data* , dans le fichier représenté par la ressource *fd* .

8.27 Accès aux dossiers

8.27.1 Introduction

8.27.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.27.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.27.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.27.5 Types de ressources

8.27.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

DIRECTORY_SEPARATOR (string)
PATH_SEPARATOR (string)

Note

La constante PATH_SEPARATOR a été introduite en PHP 4.3.0-RC2.
--

8.27.7 Voir aussi

Pour des fonctions connexes telles que [dirname](#) , [is_dir](#) , [mkdir](#) et [rmdir](#) , voyez donc la section [système de fichiers](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Voir aussi](#)
- [chdir](#)
- [chroot](#)
- [dir](#)
- [closedir](#)
- [getcwd](#)

- [opendir](#)
- [readdir](#)
- [rewinddir](#)
- [scandir](#)

8.27.9 chroot() : Change le dossier racine

bool **chroot** (string directory)

[chroot](#) change la racine du script en cours, et la remplace par directory .

Cette fonction n'est disponible que si votre système la supporte et que vous utilisez le SAPI CLI, CGI ou Embed SAPI. De plus, cette fonction nécessite les privilèges d'administrateur.

8.27.10 dir() : Classe de dossier DIR

dir **dir** string directory string path resource handle string **read** (void) void **rewind** (void) void **close** (void)

Un mécanisme pseudo-objet permet la lecture d'un dossier. L'argument directory est ouvert. Deux propriétés sont disponibles une fois le dossier ouvert : le pointeur peut être utilisé avec d'autres fonctions telles que [readdir](#) , [rewinddir](#) et [closedir](#) . Le chemin du dossier est le chemin fourni lors de la construction de l'objet. Trois méthodes sont disponibles : la lecture, la remise à zéro et la fermeture de dossier.

8.27.11 closedir() : Ferme le pointeur sur le dossier

void **closedir** (resource dir_handle)

[closedir](#) ferme le pointeur de dossier dir_handle . Le dossier devait avoir été ouvert avec [opendir](#) .

8.27.12 getcwd() : Retourne le dossier de travail courant

string **getcwd** (void)

Retourne le dossier de travail courant.

8.27.13 opendir() : Ouvre un dossier, et récupère un pointeur dessus

resource **opendir** (string path , resource context)

[opendir](#) retourne un pointeur sur un dossier qui pour être utilisé avec les fonctions [closedir](#) , [readdir](#) et [rewinddir](#) .

8.27.14 readdir() : Lit une entrée du dossier

string **readdir** (resource dir_handle)

readdir retourne le nom du fichier suivant dans le dossier identifié par dir_handle . Les noms sont retournés dans l'ordre qu'ils sont enregistrés dans le système de fichiers.

8.27.15 rewinddir() : Retourne à la première entrée du dossier

void **rewinddir** (resource dir_handle)

rewinddir retourne à la première entrée du dossier identifiée par dir_handle .

8.27.16 scandir() : Liste les fichiers et dossiers dans un dossier

array **scandir** (string directory , int sorting_order , resource context)

Retourne un tableau de fichier et dossier, issus de directory .

8.28 Fonctions DOM

8.28.1 Introduction

L'extension DOM est le remplacement de l'extension DOM XML en PHP 4. Cette extension contient encore plusieurs anciennes fonctions mais elles ne doivent plus être utilisées. En particulier, les fonctions procédurales.

Cette extension vous permet de manipuler des documents XML avec l'API DOM.

8.28.2 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.28.3 Classes pré-définies

L'API de ce module suit, autant que possible, les standards niveau 2 de DOM. En conséquent, l'API est pleinement orientée objets. Il est bien d'avoir les standards DOM sous la main lors de l'utilisation de cette extension.

Ce module définit plusieurs classes, qui sont expliquées dans les tableaux suivants. Les classes avec un équivalent dans les standards sont nommées DOMxxx.

8.28.3.1 DOMAttr

Étend DOMNode. L'interface DOMAttr représente un attribut dans l'objet DOMElement.

8.28.3.1.1 Constructeur

- DOMAttr->__construct() - construit un nouvel objet DOMAttr

8.28.3.1.2 Méthodes

- DOMAttr->isId() - Vérifie si un attribut est un ID défini

8.28.3.1.3 Propriétés

Nom	Type	Lecture seule	Description
name	string	oui	Le nom de l'attribut
ownerElement	DOMElement	oui	L'élément qui contient l'attribut
schemaTypeInfo	bool	oui	Pas encore implémenté, retourne toujours NULL
specified	bool	oui	Pas encore implémenté, retourne toujours NULL
value	string	non	La valeur de l'attribut

8.28.3.2 DOMCharacterData

Étend DOMNode .

8.28.3.2.1 Méthodes

- [DOMCharacterData->appendData\(\)](#) - Ajoute une chaîne à la fin des données du noeud
- [DOMCharacterData->deleteData\(\)](#) - Efface une rangée de caractères du noeud
- [DOMCharacterData->insertData\(\)](#) - Insère une chaîne à une position 16-bit spécifiée
- [DOMCharacterData->replaceData\(\)](#) - Remplace une sous-chaîne d'un noeud DOMCharacterData
- [DOMCharacterData->substringData\(\)](#) - Extrait une rangée de données depuis le noeud

8.28.3.2.2 Propriétés

Nom	Type	Lecture seule	Description
data	string	non	Le contenu du noeud
length	int	oui	La longueur du contenu

8.28.3.3 DOMComment

Étend DOMCharacterData .

8.28.3.3.1 Constructeur

- [DOMComment->__construct\(\)](#) - construit un nouvel objet DOMComment

8.28.3.4 DOMDocument

Étend DOMNode .

8.28.3.4.1 Constructeur

- [DOMDocument->__construct\(\)](#) - construit un nouvel objet DOMDocument

8.28.3.4.2 Méthodes

- [DOMDocument->createAttribute\(\)](#) - Crée un nouvel attribut
- [DOMDocument->createAttributeNS\(\)](#) - Crée un nouvel noeud d'attribut avec un espace de nom associé
- [DOMDocument->createCDATASection\(\)](#) - Crée un nouveau noeud cdata
- [DOMDocument->createComment\(\)](#) - Crée un nouveau noeud de commentaire
- [DOMDocument->createDocumentFragment\(\)](#) - Crée un nouveau fragment de document
- [DOMDocument->createElement\(\)](#) - Crée un nouvel élément
- [DOMDocument->createElementNS\(\)](#) - Crée un nouvel élément avec un espace de noms associé
- [DOMDocument->createEntityReference\(\)](#) - Crée une nouvelle entité de référence
- [DOMDocument->createProcessingInstruction\(\)](#) - Crée un nouveau noeud PI
- [DOMDocument->createTextNode\(\)](#) - Crée un nouveau noeud de texte

- [DOMDocument->getElementById\(\)](#) - Recherche un élément avec un ID précis
- [DOMDocument->getElementsByTagName\(\)](#) - Recherche tous les éléments ayant un tag portant un nom donné
- [DOMDocument->getElementsByTagNameNS\(\)](#) - Recherche tous les éléments ayant un tag portant un nom donné dans un espace de noms spécifié
- [DOMDocument->importNode\(\)](#) - Importe un noeud dans le document courant
- [DOMDocument->load\(\)](#) - Charge du XML depuis un fichier
- [DOMDocument->loadHTML\(\)](#) - Charge du HTML depuis une chaîne de caractères
- [DOMDocument->loadHTMLFile\(\)](#) - Charge du HTML depuis un fichier
- [DOMDocument->loadXML\(\)](#) - Charge du XML depuis une chaîne de caractères
- [DOMDocument->normalize\(\)](#) - Normalise un document
- [DOMDocument->relaxNGValidate\(\)](#) - Effectue une validation relaxNG sur le document
- [DOMDocument->relaxNGValidateSource\(\)](#) - Effectue une validation relaxNG sur le document
- [DOMDocument->save\(\)](#) - Sauvegarde l'arbre interne XML dans un fichier
- [DOMDocument->saveHTML\(\)](#) - Sauvegarde le document interne dans une chaîne de caractères en utilisant un formatage HTML
- [DOMDocument->saveHTMLFile\(\)](#) - Sauvegarde le document interne dans un fichier en utilisant un formatage HTML
- [DOMDocument->saveXML\(\)](#) - Sauvegarde l'arbre interne XML dans une chaîne de caractères
- [DOMDocument->schemaValidate\(\)](#) - Valide un document basé sur un schéma
- [DOMDocument->schemaValidateSource\(\)](#) - Valide un document basé sur un schéma
- [DOMDocument->validate\(\)](#) - Valide un document en se basant sur sa DTD
- [DOMDocument->xinclude\(\)](#) - Remplace les XIncludes dans un objet DOMDocument

8.28.3.4.3 Propriétés

Nom	Type	Lecture seule	Description
actualEncoding	string	oui	
config	DOMConfiguration	oui	
doctype	DOMDocumentType	oui	La déclaration de type de document associée avec ce document.
documentElement	DOMElement	oui	C'est un attribut de convenance qui permet l'accès direct à l'enfant d'un noeud qui est l'élément document de ce document.
documentURI	string	non	La localisation du document ou NULL si non défini.
encoding	string	non	
formatOutput	bool	non	
implementation	DOMImplementation	oui	L'objet DOMImplementation qui représente ce document.
preserveWhiteSpace	bool	non	Ne pas supprimer les espaces redondants. Vaut TRUE par défaut.
recover	bool	non	
resolveExternals	bool	non	Définir à TRUE pour charger les entités externes depuis une déclaration doctype. Ceci est utile pour inclure des entités correspondants à des caractères dans vos documents XML.

standalone	bool	non	
strictErrorChecking	bool	non	Envoi une DOMException en cas d'erreurs. Vaut par défaut TRUE .
substituteEntities	bool	non	
validateOnParse	bool	non	Charge et valide à nouveau la DTD. Vaut par défaut FALSE .
version	string	non	
xmlEncoding	string	oui	Un attribut spécifiant, en tant que partie de la déclaration XML, l'encodage de ce document. Il vaut NULL lorsqu'il n'est pas spécifié ou lorsqu'il n'est pas connu, par exemple, lorsque le document a été créé en mémoire.
xmlStandalone	bool	non	Un attribut spécifiant, en tant que partie de la déclaration XML, si oui ou non le document est autonome. Il vaut FALSE si non spécifié.
xmlVersion	string	non	Un attribut spécifiant, en tant que partie de la déclaration XML, le numéro de version de ce document. S'il n'y a pas de déclaration et si le document supporte la fonctionnalité "XML", la valeur sera "1.0".

8.28.3.5 DOMDocumentType

Étend DOMNode

Chaque DOMDocument a un attribut doctype de valeur soit NULL , soit un objet DOMDocumentType .

8.28.3.5.1 Propriétés

Nom	Type	Lecture seule	Description
publicId	string	oui	L'identifiant public du sous-ensemble externe.
systemId	string	oui	L'identifiant système du sous-ensemble externe. Il peut être une URI absolue ou non.
name	string	oui	Le nom de la DTD ; i.e. le nom suivant immédiatement le mot clé DOCTYPE
entities	DOMNamedNodeMap	oui	Un DOMNamedNodeMap contenant les entités générales, externes et internes, déclarées dans la DTD.
notations	DOMNamedNodeMap	oui	Un DOMNamedNodeMap contenant les notations déclarées dans la DTD.
internalSubset	string	oui	Le sous-ensemble interne en tant que chaîne de caractères ou NULL s'il n'y en a pas. Ceci ne contient pas les crochets délimitants.

8.28.3.6 DOMElement

Étend DOMNode .

8.28.3.6.1 Constructeur

- DOMElement->__construct() - construit un nouvel objet DOMElement

8.28.3.6.2 Méthodes

- DOMElement->getAttribute() - Retourne la valeur d'un attribut
- DOMElement->getAttributeNode() - Retourne le noeud d'un attribut
- DOMElement->getAttributeNodeNS() - Retourne le noeud d'un attribut
- DOMElement->getAttributeNS() - Retourne la valeur d'un attribut
- DOMElement->getElementsByTagName() - Récupère les éléments par le nom de leurs tags
- DOMElement->getElementsByTagNameNS() - Récupère les éléments par leur namespaceURI et leur localName
- DOMElement->hasAttribute() - Vérifie si un attribut existe
- DOMElement->hasAttributeNS() - Vérifie si un attribut existe
- DOMElement->removeAttribute() - Supprime un attribut
- DOMElement->removeAttributeNode() - Supprime un attribut
- DOMElement->removeAttributeNS() - Supprime un attribut
- DOMElement->setAttribute() - Ajoute un nouvel attribut
- DOMElement->setAttributeNode() - Ajoute un nouvel attribut à un élément
- DOMElement->setAttributeNodeNS() - Ajoute un nouvel attribut à un élément
- DOMElement->setAttributeNS() - Ajoute un nouvel attribut

8.28.3.6.3 Propriétés

Nom	Type	Lecture seule	Description
schemaTypeInfo	bool	oui	Pas encore implémenté, retourne toujours NULL
tagName	string	oui	Le nom de l'élément

8.28.3.7 DOMEntity

Étend DOMNode

Cette interface représente une entité connue, analysée ou non, du document XML.

8.28.3.7.1 Propriétés

Nom	Type	Lecture seule	Description
publicId	string	oui	L'identifiant public associé à l'entité si spécifiée, NULL sinon.
systemId	string	oui	L'identifiant système associé à l'entité si spécifié, NULL sinon. Ceci peut être une URI absolue ou non.
notationName	string	oui	Pour les entités non-analysées, le nom de la notation pour cette entité. Pour les entités analysées, ceci vaut NULL .
actualEncoding	string	non	Un attribut spécifiant l'encodage utilisé pour cette entité au moment de l'analyse, lorsque c'est une entité analysée externe. Ceci vaut NULL si c'est une entité du sous-ensemble externe ou si elle n'est pas connue.

encoding	string	oui	Un attribut spécifiant, en tant que partie de la déclaration texte, l'encodage de cette entité, lorsque c'est une entité analysée externe. Ceci vaut NULL sinon.
version	string	oui	Un attribut spécifiant, en tant que partie de la déclaration texte, le numéro de version de cette entité, lorsque c'est une entité analysée externe. Ceci vaut NULL sinon.

8.28.3.8 DOMEntityReference

Étend DOMNode .

8.28.3.8.1 Constructeur

- DOMEntityReference->__construct() - construit un nouvel objet DOMEntityReference

8.28.3.9 DOMException

Les opérations DOM lancent des exceptions sous des circonstances particulières, i.e., lorsqu'une opération est impossible à effectuer pour des raisons de logique.

Voir aussi [Exceptions](#) .

8.28.3.9.1 Propriétés

Nom	Type	Lecture seule	Description
code	int	oui	Un entier indiquant le type d'erreur générée

8.28.3.10 DOMImplementation

L'interface DOMImplementation fournit un nombre de méthodes pour effectuer des opérations qui sont indépendant de toute instance du model objet du document.

8.28.3.10.1 Constructeur

- DOMImplementation->__construct() - construit un nouvel objet DOMImplementation

8.28.3.10.2 Méthodes

- DOMImplementation->createDocument() - Crée un objet DOM Document du type spécifié avec ces éléments
- DOMImplementation->createDocumentType() - Crée un objet DOMDocumentType vide
- DOMImplementation->hasFeature() - Test si l'implémentation DOM implémente une fonctionnalité spécifique

8.28.3.11 DOMNode**8.28.3.11.1 Méthodes**

- DOMNode->appendChild() - Ajoute un nouvel enfant à la fin des enfants
- DOMNode->cloneNode() - Clone un noeud
- DOMNode->hasAttributes() - Vérifie si un noeud possède des attributs
- DOMNode->hasChildNodes() - Vérifie si un noeud possède des enfants
- DOMNode->insertBefore() - Ajoute un nouvel enfant juste avant un noeud référencé
- DOMNode->isSameNode() - Indique si deux noeuds sont identiques
- DOMNode->isSupported() - Vérifie si une fonctionnalité est supporté par une version spécifique
- DOMNode->lookupNamespaceURI() - Retourne l'URI de l'espace de noms du noeud basé sur le préfixe
- DOMNode->lookupPrefix() - Retourne le préfixe de l'espace de noms d'un noeud, basé sur l'URI de l'espace de noms
- DOMNode->normalize() - Normalise le noeud
- DOMNode->removeChild() - Supprime un enfant de la liste des enfants
- DOMNode->replaceChild() - Remplace un enfant

8.28.3.11.2 Propriétés

Nom	Type	Lecture seule	Description
nodeName	string	oui	Retourne le nom le plus précis possible pour le type de noeud courant
nodeValue	string	non	La valeur de ce noeud, dépendant de son type.
nodeType	int	oui	Récupère le type du noeud. Une des constantes XML_XXX_NODE prédéfinies
parentNode	DOMNode	oui	Le parent de ce noeud
childNodes	DOMNodeList	oui	Un DOMNodeList qui contient tous les enfants de ce noeud. S'il n'y a pas d'enfant, ce sera un DOMNodeList vide.
firstChild	DOMNode	oui	Le premier enfant de ce noeud. S'il n'y pas de tel noeud, ceci retourne NULL .
lastChild	DOMNode	oui	Le dernier enfant de ce noeud. S'il n'y a pas de tel noeud, ceci retourne NULL .
previousSibling	DOMNode	oui	Le noeud précédent immédiatement ce noeud. S'il n'y a pas de tel noeud, ceci retourne NULL .
nextSibling	DOMNode	oui	Le noeud suivant immédiatement ce noeud. S'il n'y a pas de tel noeud, ceci retourne NULL .

attributes	DOMNamedNodeMap	oui	Un DOMNamedNodeMap contenant les attributs de ce noeud (si c'est un DOMElement) ou NULL sinon.
ownerDocument	DOMDocument	oui	Un objet DOMDocument associé à ce noeud.
namespaceURI	string	oui	L'URI de l'espace de noms de ce noeud, ou NULL si non spécifié.
prefix	string	non	Le préfixe de l'espace de noms de ce noeud, ou NULL si non spécifié.
localName	string	oui	Retourne la partie locale du nom qualifié de ce noeud.
baseURI	string	oui	La base absolue de l'URI de ce noeud ou NULL si l'implémentation n'est pas arrivée à obtenir une URI absolue.
textContent	string	non	Cet attribut retourne le contenu texte de ce noeud et ces descendants.

8.28.3.12 DOMNodeList

8.28.3.12.1 Méthodes

- DOMNodeList->item() - Retrieves a node specified by index

8.28.3.12.2 Propriétés

Nom	Type	Lecture seule	Description
length	int	oui	Le nombre de noeuds dans la liste. La plage des indices des noeuds enfants valides est de 0 à length - 1 inclusivement.

8.28.3.13 DOMNotation

Étend DOMNode

8.28.3.13.1 Propriétés

Nom	Type	Lecture seule	Description
publicId	string	oui	
systemId	string	oui	

8.28.3.14 DOMProcessingInstruction

Étend DOMNode .

8.28.3.14.1 Constructeur

- DOMProcessingInstruction->__construct() - construit un nouvel objet DOMProcessingInstruction

8.28.3.14.2 Propriétés

Nom	Type	Lecture seule	Description
target	string	oui	
data	string	non	

8.28.3.15 DOMText

Étend DOMCharacterData .

8.28.3.15.1 Constructeur

- DOMText->__construct() - construit un nouvel objet DOMText

8.28.3.15.2 Méthodes

- DOMText->isWhitespaceInElementContent() - Indique si ce noeud de texte contient des espaces
- DOMText->splitText() - Sépare le noeud en deux noeuds à une position spécifique

8.28.3.15.3 Propriétés

Nom	Type	Lecture seule	Description
wholeText	string	oui	

8.28.3.16 DOMXPath

8.28.3.16.1 Constructeur

- DOMXPath->__construct() - Construit un nouvel objet DOMXPath

8.28.3.16.2 Méthodes

- DOMXPath->registerNamespace() - Enregistre l'espace de noms avec l'objet DOMXPath
- DOMXPath->evaluate() - Évalue l'expression XPath donnée et retourne le résultat écrit
- DOMXPath->query() - Évalue l'expression XPath donnée

8.28.3.16.3 Propriétés

Nom	Type	Lecture seule	Description
document	DOMDocument		

8.28.4 Exemples

La plupart des exemples de cette référence requièrent un fichier XML. Nous utiliserons le fichier book.xml qui contient ce qui suit :

chapter.xml
<pre> <!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN" "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd" []> <book id="listing"> <title>My lists</title> <chapter id="books"> <title>My books</title> <para> <informaltable> <tgroup cols="4"> <thead> <row> <entry>Title</entry> <entry>Author</entry> <entry>Language</entry> <entry>ISBN</entry> </row> </thead> <tbody> <row> <entry>The Grapes of Wrath</entry> <entry>John Steinbeck</entry> <entry>en</entry> <entry>0140186409</entry> </row> <row> <entry>The Pearl</entry> <entry>John Steinbeck</entry> <entry>en</entry> <entry>014017737X</entry> </row> <row> <entry>Samarcande</entry> <entry>Amine Maalouf</entry> <entry>fr</entry> <entry>2253051209</entry> </row> </tbody> </tgroup> </informaltable> </para> </chapter> </book> </pre>

8.28.5 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Constante	Valeur	Description
XML_ELEMENT_NODE (<u>entier</u>)	1	Le noeud est un DOMElement
XML_ATTRIBUTE_NODE (<u>entier</u>)	2	Le noeud est un DOMAttr
XML_TEXT_NODE (<u>entier</u>)	3	Le noeud est un DOMText
XML_CDATA_SECTION_NODE (<u>entier</u>)	4	Le noeud est un DOMCharacterData
XML_ENTITY_REF_NODE (<u>entier</u>)	5	Le noeud est un DOMEntityReference
XML_ENTITY_NODE (<u>entier</u>)	6	Le noeud est un DOMEntity
XML_PI_NODE (<u>entier</u>)	7	Le noeud est un DOMProcessingInstruction
XML_COMMENT_NODE (<u>entier</u>)	8	Le noeud est un DOMComment
XML_DOCUMENT_NODE (<u>entier</u>)	9	Le noeud est un DOMDocument
XML_DOCUMENT_TYPE_NODE (<u>entier</u>)	10	Le noeud est un DOMDocumentType
XML_DOCUMENT_FRAG_NODE (<u>entier</u>)	11	Le noeud est un DOMDocumentFragment
XML_NOTATION_NODE (<u>entier</u>)	12	Le noeud est un DOMNotation
XML_HTML_DOCUMENT_NODE (<u>entier</u>)	13	
XML_DTD_NODE (<u>entier</u>)	14	
XML_ELEMENT_DECL_NODE (<u>entier</u>)	15	
XML_ATTRIBUTE_DECL_NODE (<u>entier</u>)	16	
XML_ENTITY_DECL_NODE (<u>entier</u>)	17	
XML_NAMESPACE_DECL_NODE (<u>entier</u>)	18	
XML_ATTRIBUTE_CDATA (<u>entier</u>)	1	
XML_ATTRIBUTE_ID (<u>entier</u>)	2	
XML_ATTRIBUTE_IDREF (<u>entier</u>)	3	
XML_ATTRIBUTE_IDREFS (<u>entier</u>)	4	
XML_ATTRIBUTE_ENTITY (<u>entier</u>)	5	
XML_ATTRIBUTE_NMTOKEN (<u>entier</u>)	7	
XML_ATTRIBUTE_NMTOKENS (<u>entier</u>)	8	
XML_ATTRIBUTE_ENUMERATION (<u>entier</u>)	9	
XML_ATTRIBUTE_NOTATION (<u>entier</u>)	10	
Constante	Valeur	Description
DOM_INDEX_SIZE_ERR (<u>entier</u>)	1	Si l'index ou la taille est négatif ou plus grand que la valeur autorisée.
DOMSTRING_SIZE_ERR (<u>entier</u>)	2	Si la rangée spécifiée de texte ne tient pas dans le DOMString .
DOM_HIERARCHY_REQUEST_ERR (<u>entier</u>)	3	Si un noeud est inséré à un endroit non autorisé
DOM_WRONG_DOCUMENT_ERR (<u>entier</u>)	4	Si un noeud est utilisé dans un document autre que celui qui l'a créé.
DOM_INVALID_CHARACTER_ERR (<u>entier</u>)	5	Si un caractère invalide ou non autorisé est spécifié, par exemple, dans un nom.
DOM_NO_DATA_ALLOWED_ERR (<u>entier</u>)	6	Si des données sont spécifiées dans un noeud qui ne supporte pas les données.
DOM_NO_MODIFICATION_ALLOWED_ERR (<u>entier</u>)	7	Si une tentative est faite de modifier un objet où les modifications ne sont pas autorisées.
DOM_NOT_FOUND_ERR (<u>entier</u>)	8	Si une tentative est faite de référencer un noeud dans un contexte où il n'existe pas.

DOM_NOT_SUPPORTED_ERR (entier)	9	Si l'implémentation ne supporte pas le type requis de l'objet ou de l'opération.
DOM_INUSE_ATTRIBUTE_ERR (entier)	10	Si une tentative est faite d'ajouter un attribut qui est déjà utilisé autre part.
DOM_INVALID_STATE_ERR (entier)	11	Si une tentative est faite d'utiliser un objet qui n'est pas, ou n'est plus, utilisable.
DOM_SYNTAX_ERR (entier)	12	Si une chaîne de caractères invalide ou illégale est spécifiée.
DOM_INVALID_MODIFICATION_ERR (entier)	13	Si une tentative est faite de modifier le type d'un objet fondamental.
DOM_NAMESPACE_ERR (entier)	14	Si une tentative est faite de créer ou de modifier un objet dans un sens qui est incorrect au regard de l'espace de noms.
DOM_INVALID_ACCESS_ERR (entier)	15	Si un paramètre ou une opération n'est pas supporté par l'objet fondamental.
DOM_VALIDATION_ERR (entier)	16	Si un appel à une méthode tel que insertBefore ou removeChild voudrait rendre le noeud invalide avec le respect de la validation partielle, cette exception sera exécutée et l'opération ne sera pas effectuée.

- [Introduction](#)
- [Installation](#)
- [Classes pré-définies](#)
- [Exemples](#)
- [Constantes pré-définies](#)
- [DOMAttr->__construct\(\)](#)
- [DOMAttr->isId\(\)](#)
- [DOMCharacterData->appendData\(\)](#)
- [DOMCharacterData->deleteData\(\)](#)
- [DOMCharacterData->insertData\(\)](#)
- [DOMCharacterData->replaceData\(\)](#)
- [DOMCharacterData->substringData\(\)](#)
- [DOMComment->__construct\(\)](#)
- [DOMDocument->__construct\(\)](#)
- [DOMDocument->createAttribute\(\)](#)
- [DOMDocument->createAttributeNS\(\)](#)
- [DOMDocument->createCDATASection\(\)](#)
- [DOMDocument->createComment\(\)](#)
- [DOMDocument->createDocumentFragment\(\)](#)
- [DOMDocument->createElement\(\)](#)
- [DOMDocument->createElementNS\(\)](#)
- [DOMDocument->createEntityReference\(\)](#)
- [DOMDocument->createProcessingInstruction\(\)](#)
- [DOMDocument->createTextNode\(\)](#)
- [DOMDocument->getElementById\(\)](#)
- [DOMDocument->getElementsByTagName\(\)](#)
- [DOMDocument->getElementsByTagNameNS\(\)](#)
- [DOMDocument->importNode\(\)](#)
- [DOMDocument->load\(\)](#)
- [DOMDocument->loadHTML\(\)](#)
- [DOMDocument->loadHTMLFile\(\)](#)

- DOMDocument->loadXML()
- DOMDocument->normalize()
- DOMDocument->relaxNGValidate()
- DOMDocument->relaxNGValidateSource()
- DOMDocument->save()
- DOMDocument->saveHTML()
- DOMDocument->saveHTMLFile()
- DOMDocument->saveXML()
- DOMDocument->schemaValidate()
- DOMDocument->schemaValidateSource()
- DOMDocument->validate()
- DOMDocument->xinclude()
- DOMElement->__construct()
- DOMElement->getAttribute()
- DOMElement->getAttributeNode()
- DOMElement->getAttributeNodeNS()
- DOMElement->getAttributeNS()
- DOMElement->getElementsByTagName()
- DOMElement->getElementsByTagNameNS()
- DOMElement->hasAttribute()
- DOMElement->hasAttributeNS()
- DOMElement->removeAttribute()
- DOMElement->removeAttributeNode()
- DOMElement->removeAttributeNS()
- DOMElement->setAttribute()
- DOMElement->setAttributeNode()
- DOMElement->setAttributeNodeNS()
- DOMElement->setAttributeNS()
- DOMEntityReference->__construct()
- DOMImplementation->__construct()
- DOMImplementation->createDocument()
- DOMImplementation->createDocumentType()
- DOMImplementation->hasFeature()
- DOMNamedNodeMap->getNamedItem()
- DOMNamedNodeMap->getNamedItemNS()
- DOMNamedNodeMap->item()
- DOMNode->appendChild()
- DOMNode->cloneNode()
- DOMNode->hasAttributes()
- DOMNode->hasChildNodes()
- DOMNode->insertBefore()
- DOMNode->isSameNode()
- DOMNode->isSupported()
- DOMNode->lookupNamespaceURI()
- DOMNode->lookupPrefix()
- DOMNode->normalize()
- DOMNode->removeChild()
- DOMNode->replaceChild()
- DOMNodeList->item()
- DOMProcessingInstruction->__construct()
- DOMText->__construct()
- DOMText->isWhitespaceInElementContent()
- DOMText->splitText()
- DOMXPath->__construct()

- [DOMXPath->evaluate\(\)](#)
- [DOMXPath->query\(\)](#)
- [DOMXPath->registerNamespace\(\)](#)
- [dom_import_simplexml](#)

8.28.7 DOMAttr->isId(): Vérifie si l'attribut est un identifiant défini

DOMAttr bool **isId** (void)

Cette fonction vérifie si l'attribut est un identifiant défini. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

En accord avec le standard DOM, ceci requiert une DTD qui définit l'attribut ID qui est du type ID. Vous devez valider votre document avec la fonction [DOMDocument->validate\(\)](#) ou la fonction `DOMDocument::validateOnParse` avant d'utiliser cette fonction.

8.28.8 DOMCharacterData->appendData(): Ajoute la chaîne à la fin des données dans le noeud

DOMCharacterData void **appendData** (string data)

Ajoute la chaîne data à la fin des données dans le noeud.

8.28.9 DOMCharacterData->deleteData(): Supprime une sous-chaîne dans le noeud

DOMCharacterData void **deleteData** (int offset , int count)

Efface count caractères à partir de la position offset .

8.28.10 DOMCharacterData->insertData(): Insère une chaîne à la position spécifiée d'unité 16-bit

DOMCharacterData void **insertData** (int offset , string data)

Insère la chaîne data à la position offset .

8.28.11 DOMCharacterData->replaceData(): Remplace une sous-chaîne dans le noeud DOMCharacterData node

DOMCharacterData void **replaceData** (int offset , int count , string data)

Remplace count caractères en partant de la position offset par les données data .

8.28.12 DOMCharacterData->substringData()() : Extrait un morceaux de données dans le noeud

DOMCharacterData string **substringData** (int offset , int count)

Retourne la sous-chaîne spécifiée.

8.28.13 DOMComment->__construct()() : Crée un nouvel objet DOMComment

DOMComment **__construct** string value

Crée un nouvel objet DOMComment . Cet objet est en lecture seule. Il peut être ajouté à un document, mais les noeuds additionnels ne peuvent pas être ajoutés à ce noeud tant que ce noeud n'est pas associé à un document. Pour créer un noeud accessible en écriture, utilisez [DOMDocument->createComment\(\)](#) .

8.28.14 DOMDocument->__construct()() : Crée un nouvel objet DOMDocument

DOMDocument **__construct** string version string encoding

Crée un nouvel objet DOMDocument .

8.28.15 DOMDocument->createAttribute()() : Crée un nouvel attribut

DOMDocument DOMAttr **createAttribute** (string name)

Cette fonction crée une nouvelle instance de la classe DOMAttr . Ce noeud ne sera pas affiché dans le document, à moins qu'il ne soit inséré avec [DOMNode->appendChild\(\)](#) .

8.28.16 DOMDocument->createAttributeNS()() : Crée un nouvel attribut avec un espace de noms associé

DOMDocument DOMAttr **createAttributeNS** (string namespaceURI , string qualifiedName)

Cette fonction crée une nouvelle instance de la classe DOMAttr . Ce noeud ne sera pas affiché dans le document, à moins qu'il ne soit inséré avec [DOMNode->appendChild\(\)](#) .

8.28.17 DOMDocument->createCDATASection()() : Crée un nouveau noeud cdata

DOMDocument DOMCDATASection **createCDATASection** (string data)

Cette fonction crée une nouvelle instance de la classe DOMCDATASection . Ce noeud ne sera pas affiché dans le document, à moins qu'il ne soit inséré avec [DOMNode->appendChild\(\)](#) .

8.28.18 DOMDocument->createComment()() : Crée un nouveau noeud de commentaire

DOMDocument DOMComment **createComment** (string data)

Cette fonction crée une nouvelle instance de la classe DOMComment . Ce noeud ne sera pas affiché dans le document, à moins qu'il ne soit inséré avec [DOMNode->appendChild\(\)](#) .

8.28.19 DOMDocument->createDocumentFragment()() : Crée un nouveau fragment de document

DOMDocument DOMDocumentFragment **createDocumentFragment** (void)

Cette fonction crée une nouvelle instance de la classe DOMDocumentFragment . Ce noeud ne sera pas affiché dans le document, à moins qu'il ne soit inséré avec [DOMNode->appendChild\(\)](#) .

8.28.20 DOMDocument->createElement()() : Crée un nouveau noeud

DOMDocument DOMElement **createElement** (string name , string value)

Cette fonction crée une nouvelle instance de la classe DOMElement . Ce noeud ne sera pas affiché dans le document, à moins qu'il ne soit inséré avec [DOMNode->appendChild\(\)](#) .

8.28.21 DOMDocument->createElementNS()() : Crée un nouveau noeud avec un espace de noms associé

DOMDocument DOMElement **createElementNS** (string namespaceURI , string qualifiedName , string value)

Cette fonction crée un nouveau noeud avec un espace de noms associé. Ce noeud ne sera pas affiché dans le document, à moins qu'il ne soit inséré avec [DOMNode->appendChild\(\)](#) .

8.28.22 DOMDocument->createEntityReference()() : Crée un nouveau noeud de référence d'entité

DOMDocument DOMEntityReference **createEntityReference** (string name)

Cette fonction crée une nouvelle instance de la classe DOMEntityReference . Ce noeud ne sera pas affiché dans le document, à moins qu'il ne soit inséré avec [DOMNode->appendChild\(\)](#) .

8.28.23 DOMDocument->createProcessingInstruction()() : Crée un nouveau noeud PI

DOMDocument DOMProcessingInstruction **createProcessingInstruction** (string target , string data)

Cette fonction crée une nouvelle instance de la classe `DOMProcessingInstruction` . Ce noeud ne sera pas affiché dans le document, à moins qu'il ne soit inséré avec `DOMNode->appendChild()` .

8.28.24 DOMDocument->createTextNode() : Crée un nouveau noeud de texte

`DOMDocument DOMText createTextNode (string content)`

Cette fonction crée une nouvelle instance de la classe `DOMText` . Ce noeud ne sera pas affiché dans le document, à moins qu'il ne soit inséré avec `DOMNode->appendChild()` .

8.28.25 DOMDocument->getElementById() : Cherche un élément avec un certain identifiant

`DOMDocument DOMElement getElementById (string elementId)`

Cette fonction est similaire à la fonction `DOMDocument->getElementsByTagName()` mais cherche un élément avec un identifiant donné.

Selon le standard DOM, ceci requiert une DTD qui définit l'attribut ID du type ID. Vous devez valider votre document avec la fonction `DOMDocument->validate()` ou `DOMDocument::validateOnParse` avant d'utiliser cette fonction.

8.28.26 DOMDocument->getElementsByTagName() : Cherche tous les éléments qui ont le nom de balise donné

`DOMDocument DOMNodeList getElementsByTagName (string name)`

Cette fonction retourne une instance de la classe `DOMNodeList` contenant les éléments qui ont un nom de balise donné.

8.28.27 DOMDocument->getElementsByTagNameNS() : Recherche tous les éléments avec un nom de balise donné dans un espace de noms spécifié

`DOMDocument DOMNodeList getElementsByTagNameNS (string namespaceURI , string localName)`

Retourne un `DOMNodeList` de tous les éléments avec un nom local donné et une URI d'espace de noms.

8.28.28 DOMDocument->importNode() : Importe un noeud dans le document courant

`DOMDocument DOMNode importNode (DOMNode importedNode , bool deep)`

Cette fonction retourne une copie du noeud à importer et l'associe avec le document courant.

8.28.29 DOMDocument->load()() : Charge du XML depuis un fichier

DOMDocument bool **load** (string filename , int options)

Charge un document XML depuis un fichier.

Cette méthode peut également être appelée statiquement pour charger et créer un objet DOMDocument . L'invoquation statique peut être utilisée lorsqu'aucune propriété de la classe DOMDocument n'a besoin d'être définie avant le chargement.

8.28.30 DOMDocument->loadHTML()() : Charge du HTML à partir d'une chaîne de caractères

DOMDocument bool **loadHTML** (string source)

Cette fonction analyse un document HTML contenu dans la chaîne source . Contrairement au XML, le HTML n'a pas besoin d'être bien formé pour être chargé. Cette fonction peut aussi être appelée statiquement pour charger et créer un objet DOMDocument . L'appel statique peut être utilisé lorsque vous n'avez besoin de configurer aucune propriété de DOMDocument avant le chargement.

8.28.31 DOMDocument->loadHTMLFile()() : Charge du HTML à partir d'un fichier

DOMDocument bool **loadHTMLFile** (string filename)

Cette fonction analyse un document HTML contenu dans le fichier filename . Contrairement au XML, le HTML n'a pas besoin d'être bien formé pour être chargé.

Cette fonction peut aussi être appelée statiquement pour charger et créer un objet DOMDocument . L'appel statique peut être utilisé lorsque vous n'avez besoin de configurer aucune propriété de DOMDocument avant le chargement.

8.28.32 DOMDocument->loadXML()() : Charge du XML depuis une chaîne de caractères

DOMDocument bool **loadXML** (string source , int options)

Charge un document XML depuis une chaîne de caractères.

Cette méthode peut être appelée statiquement pour charger et créer un objet DOMDocument . L'invoquation statique peut être utilisée lorsqu'aucune propriété de la classe DOMDocument n'a besoin d'être définie avant le chargement.

8.28.33 DOMDocument->normalize()() : Normalise le document

DOMDocument void **normalize** (void)

Normalise le document.

8.28.34 DOMDocument->relaxNGValidate()() : Effectue une validation relaxNG sur le document

DOMDocument bool **relaxNGValidate** (string filename)

Effectue une validation relaxNG sur le document selon le schéma RNG défini par filename .

8.28.35 DOMDocument->relaxNGValidateSource()() : Effectue une validation relaxNG sur le document

DOMDocument bool **relaxNGValidateSource** (string source)

Effectue une validation relaxNG sur le document selon le schéma RNG définit dans la chaîne source .

8.28.36 DOMDocument->save()() : Sauvegarde l'arbre interne XML dans un fichier

DOMDocument mixed **save** (string filename , *integer options*)

Crée un document XML depuis la représentation DOM. Cette fonction est habituellement appelée après la construction d'un tout nouveau document DOM, comme dans l'exemple ci-dessous.

8.28.37 DOMDocument->saveHTML()() : Sauvegarde le document interne dans une chaîne en utilisant un formatage HTML

DOMDocument string **saveHTML** (void)

Crée un document HTML depuis une représentation DOM. Cette fonction est habituellement appelée après la construction d'un tout nouveau document DOM, comme dans l'exemple ci-dessous.

8.28.38 DOMDocument->saveHTMLFile()() : Sauvegarde un document interne dans un fichier en utilisant un formatage HTML

DOMDocument int **saveHTMLFile** (string filename)

Crée un document HTML depuis une représentation DOM. Cette fonction est habituellement appelée après la construction d'un tout nouveau document DOM, comme dans l'exemple ci-dessous.

8.28.39 DOMDocument->saveXML()() : Sauvegarde l'arbre interne XML dans une chaîne de caractères

DOMDocument string **saveXML** (*DOMNode node* , *integer options*)

Crée un document XML depuis la représentation DOM. Cette fonction est habituellement appelée après la création d'un tout nouveau document DOM, comme dans l'exemple ci-dessous.

8.28.40 DOMDocument->schemaValidate()() : Valide un document selon un schéma

DOMDocument bool **schemaValidate** (string filename)

Valide un document selon un schéma contenu dans un fichier donné.

8.28.41 DOMDocument->schemaValidateSource()() : Valide un document selon un schéma

DOMDocument bool **schemaValidateSource** (string source)

Valide un document selon un schéma défini dans la chaîne donnée.

8.28.42 DOMDocument->validate()() : Valide un document en se basant sur sa DTD

DOMDocument bool **validate** (void)

Valide un document en se basant sur sa DTD.

Vous pouvez utiliser la propriété `validateOnParse` de la classe `DOMDocument` pour effectuer une validation DTD.

8.28.43 DOMDocument->xinclude()() : Remplace les XIncludes dans un objet DOMDocument

DOMDocument int **xinclude** (*int options*)

Cette méthode remplace les XIncludes dans un objet `DOMDocument`.

Note
Vu que la bibliothèque libxml2 résout automatiquement les entités, cette méthode peut produire des résultats non attendus si le fichier XML inclus a une DTD d'attachée.

8.28.44 DOMELEMENT->__construct()() : Crée un nouvel objet DOMELEMENT

DOMELEMENT **__construct** string name string value string namespaceURI

Crée un nouvel objet `DOMElement` . Cet objet est en lecture seule. Il peut être ajouté à un document, mais les noeuds additionnels ne peuvent pas être ajoutés à ce noeud tant que ce noeud est associé à un document. Pour créer un noeud accessible en écriture, utilisez `DOMDocument->createElement()` ou `DOMDocument->createElementNS()` .

8.28.45 `DOMElement->getAttribute()` : Retourne la valeur d'un attribut

`DOMElement` string **`getAttribute`** (string name)

Récupère la valeur d'un attribut avec le nom `name` pour le noeud courant.

8.28.46 `DOMElement->getAttributeNode()` : Retourne le noeud d'attribut

`DOMElement` `DOMAttr` **`getAttributeNode`** (string name)

Retourne le noeud d'attribut avec le nom `name` pour l'élément courant.

8.28.47 `DOMElement->getAttributeNodeNS()` : Retourne le noeud d'attribut

`DOMElement` `DOMAttr` **`getAttributeNodeNS`** (string namespaceURI , string localName)

Retourne le noeud d'attribut dans l'espace de noms `namespaceURI` avec le nom local `localName` pour le noeud courant.

8.28.48 `DOMElement->getAttributeNS()` : Retourne la valeur de l'attribut

`DOMElement` string **`getAttributeNS`** (string namespaceURI , string localName)

Récupère la valeur de l'attribut dans l'espace de noms `namespaceURI` avec le nom local `localName` pour le noeud courant.

8.28.49 `DOMElement->getElementsByTagName()` : Retourne les éléments par leur nom de balise

`DOMElement` `DOMNodeList` **`getElementsByTagName`** (string name)

Cette fonction retourne une nouvelle instance de la classe `DOMNodeList` contenant tous les éléments descendants avec un nom de balise `name` donné, dans l'ordre dans lesquels ils sont rencontrés lors du parcours de l'arbre de cet élément.

8.28.50 `DOMElement->getElementsByTagNameNS()` : Récupère les éléments par leur espace de noms et leur `localName`

`DOMElement` `DOMNodeList` **`getElementsByTagNameNS`** (string namespaceURI , string localName)

Cette fonction récupère tous les éléments descendants avec un nom local `localName` et un espace de noms `namespaceURI` donnés.

8.28.51 `DOMElement->hasAttribute()` : Vérifie si un attribut existe

`DOMElement` bool **`hasAttribute`** (string `name`)

Indique si un attribut nommé `name` existe en tant que membre de l'élément.

8.28.52 `DOMElement->hasAttributeNS()` : Vérifie si un attribut existe

`DOMElement` bool **`hasAttributeNS`** (string `namespaceURI` , string `localName`)

Vérifie si un attribut dans l'espace de noms `namespaceURI` nommé `localName` existe en tant que membre de l'élément.

8.28.53 `DOMElement->removeAttribute()` : Efface un attribut

`DOMElement` bool **`removeAttribute`** (string `name`)

Efface l'attribut nommé `name` de l'élément.

8.28.54 `DOMElement->removeAttributeNode()` : Efface un attribut

`DOMElement` bool **`removeAttributeNode`** (`DOMAttr` `oldnode`)

Efface l'attribut `oldnode` de l'élément.

8.28.55 `DOMElement->removeAttributeNS()` : Efface un attribut

`DOMElement` bool **`removeAttributeNS`** (string `namespaceURI` , string `localName`)

Efface un attribut avec l'espace de noms `namespaceURI` nommé `localName` de l'élément.

8.28.56 `DOMElement->setAttribute()` : Ajoute un nouvel attribut

`DOMElement` bool **`setAttribute`** (string `name` , string `value`)

Ajoute un attribut avec `name` comme nom `name` et comme valeur `value` . Si l'attribut n'existe pas, il sera créé.

8.28.57 `DOMElement->setAttributeNode()` : Ajoute un nouvel attribut à l'élément

`DOMElement` `DOMAttr` **`setAttributeNode`** (`DOMAttr` `attr`)

8.28.50 `DOMElement->getElementsByTagNameNS()` : Récupère les éléments par leur espace de noms et

Ajoute un nouvel attribut `attr` à l'élément.

8.28.58 DOMElement->setAttributeNodeNS() : Ajoute un nouvel attribut à l'élément

DOMElement DOMAttr **setAttributeNodeNS** (DOMAttr `attr`)

Ajoute un nouvel attribut `attr` à l'élément.

8.28.59 DOMElement->setAttributeNS() : Ajoute un nouvel attribut

DOMElement void **setAttributeNS** (string `namespaceURI` , string `qualifiedName` , string `value`)

Ajoute un attribut avec `namespaceURI` comme espace de noms et `qualifiedName` comme nom et `value` comme valeur. Si l'attribut n'existe pas, il sera créé.

8.28.60 DOMEntityReference->__construct() : Crée un nouvel objet DOMEntityReference

DOMEntityReference **__construct** string `name`

Crée un nouvel objet `DOMEntityReference` .

8.28.61 DOMImplementation->__construct() : Crée un nouvel objet DOMImplementation

DOMImplementation **__construct** void

Crée un nouvel objet `DOMImplementation` .

8.28.62 DOMImplementation->createDocument() : Crée un objet DOM Document du type spécifié avec ses éléments

DOMImplementation DOMDocument **createDocument** (string `namespaceURI` , string `qualifiedName` , `DOMDocumentType` `doctype`)

Crée un objet `DOMDocument` du type spécifié avec ces éléments du document.

8.28.63 DOMImplementation->createDocumentType() : Crée un objet DOMDocumentType vide

DOMImplementation DOMDocumentType **createDocumentType** (string `qualifiedName` , string `publicId` , string `systemId`)

Crée un objet `DOMDocumentType` vide. Les déclarations et notations d'entités ne sont pas disponibles. Les expansions de références d'entités et les ajouts d'attributs par défaut ne sont pas non plus effectuées.

8.28.64 DOMImplementation->hasFeature()() : Test si l'implémentation DOM implémente une fonctionnalité spécifique

`DOMImplementation` bool **hasFeature** (string feature , string version)

Test si l'implémentation DOM implémente une fonctionnalité feature spécifique.

Vous pouvez trouver une liste de toutes les fonctionnalités dans la section [Conformance](#) de la spécification DOM.

8.28.65 DOMNamedNodeMap->getNamedItem()() : Retourne un noeud spécifié par nom nom

`DOMNamedNodeMap` `DOMNode` **getNamedItem** (string name)

Récupère un noeud spécifié par son nodeName .

8.28.66 DOMNamedNodeMap->getNamedItemNS()() : Retourne un noeud spécifié par son nom local et son espace de nom

`DOMNamedNodeMap` `DOMNode` **getNamedItemNS** (string namespaceURI , string localName)

Retourne un noeud spécifié par localName et namespaceURI .

8.28.67 DOMNamedNodeMap->item()() : Retourne un noeud selon son index

`DOMNamedNodeMap` `DOMNode` **item** (int index)

Récupère un noeud spécifié par index dans l'objet `DOMNamedNodeMap` .

8.28.68 DOMNode->appendChild()() : Ajoute un nouveau fils à la fin des fils

`DOMNode` `DOMNode` **appendChild** (`DOMNode` newnode)

Cette fonction ajoute un fils à une liste de fils existant ou crée une nouvelle liste de fils. Le fils peut être créé avec, e.g. [DOMDocument->createElement\(\)](#) , [DOMDocument->createTextNode\(\)](#) etc. ou simplement en utilisant tout autre noeud.

8.28.69 DOMNode->cloneNode()() : Clone un noeud

DOMNode DOMNode **cloneNode** (*bool* *deep*)

Crée une copie d'un noeud.

8.28.70 DOMNode->hasAttributes()() : Vérifie si le noeud possède un attribut

DOMNode bool **hasAttributes** (void)

Cette méthode vérifie si un noeud possède des attributs. Le noeud testé doit être un XML_ELEMENT_NODE .

8.28.71 DOMNode->hasChildNodes()() : Vérifie si le noeud possède des enfants

DOMNode bool **hasChildNodes** (void)

Cette fonction vérifie si un noeud possède des enfants.

8.28.72 DOMNode->insertBefore()() : Ajoute un nouveau fils à la fin des enfants

DOMNode DOMNode **insertBefore** (DOMNode *newnode* , *DOMNode* *refnode*)

Cette fonction insère un nouveau noeud juste avec le noeud de référence. Si vous comptez faire des modifications plus tard sur l'enfant ajouté, vous devez utiliser le noeud retourné.

8.28.73 DOMNode->isSameNode()() : Indique si deux noeuds sont identiques

DOMNode bool **isSameNode** (DOMNode *node*)

Cette fonction indique si deux noeuds sont identiques. La comparaison n'est **pas** basée sur le contenu.

8.28.74 DOMNode->isSupported()() : Vérifie si la fonctionnalité est disponible pour la version spécifiée

DOMNode bool **isSupported** (string *feature* , string *version*)

Vérifie si la fonctionnalité *feature* demandée est supportée par la version *version* spécifiée.

8.28.75 DOMNode->lookupNamespaceURI()() : Retourne l'URI de l'espace de noms selon le préfixe

DOMNode string **lookupNamespaceURI** (string prefix)

Retourne l'URI de l'espace de noms selon le préfixe prefix .

8.28.76 DOMNode->lookupPrefix()() : Retourne le préfixe de l'espace de noms selon l'URI de l'espace de noms

DOMNode string **lookupPrefix** (string namespaceURI)

Retourne le préfixe de l'espace du noeud selon l'URI de l'espace de noms.

8.28.77 DOMNode->normalize()() : Normalise le noeud

DOMNode void **normalize** (void)

Normalise le noeud.

8.28.78 DOMNode->removeChild()() : Supprime un fils de la liste des enfants

DOMNode DOMNode **removeChild** (DOMNode oldnode)

Cette fonction supprime un fils de la liste des enfants.

8.28.79 DOMNode->replaceChild()() : Remplace un fils

DOMNode DOMNode **replaceChild** (DOMNode newnode , DOMNode oldnode)

Cette fonction remplace le fils oldnode par le nouveau noeud spécifié. Si le nouveau noeud est déjà un fils, il ne sera pas ajouter une seconde fois. Si le remplacement réussi, l'ancien noeud sera retourné.

8.28.80 DOMNodelist->item()() : Retourne un noeud spécifié par son index

DOMNodeList DOMNode **item** (int index)

Retourne un noeud spécifié par son index dans l'objet DOMNodeList .

Array

8.28.81 DOMProcessingInstruction->__construct()() : Crée un nouvel objet DOMProcessingInstruction

DOMProcessingInstruction **__construct** string name string value

Crée un nouvel objet DOMProcessingInstruction . Cet objet est en lecture seule. Il peut être ajouté à un document, mais les noeuds additionnels ne peuvent pas être ajoutés à ce noeud tant que ce noeud est associé à un document. Pour créer un noeud accessible en écriture, utilisez [DOMDocument->createProcessingInstruction\(\)](#) .

8.28.82 DOMText->__construct()() : Crée un nouvel objet DOMText

DOMText **__construct** string value

Crée un nouvel objet DOMText .

8.28.83 DOMText->isWhitespaceInElementContent()() : Indique si ce noeud de texte contient des espaces blancs

DOMText bool **isWhitespaceInElementContent** (void)

Indique si un noeud de texte contient des espaces. Cette détermination est faite durant le chargement du document.

8.28.84 DOMText->splitText()() : Coupe le noeud en deux noeuds à l'endroit spécifié

DOMText DOMText **splitText** (int offset)

Coupe le noeud en deux noeuds à l'endroit spécifié offset , en gardant les deux dans l'arbre comme siblings.

Après la césure, ce noeud contiendra tout le texte jusqu'à offset . Un nouveau noeud avec le reste du texte sera retourné. Si le noeud d'origine possède un parent, le nouveau noeud est inséré à la suite du noeud courant. Lorsque offset est égal à la taille du noeud, le nouveau noeud est vide.

8.28.85 DOMXPath->__construct()() : Crée un nouvel objet DOMXPath

DOMXPath **__construct** DOMDocument doc

Crée un nouvel objet DOMXPath .

8.28.86 DOMXPath->evaluate()() : Evalue une expression XPath donnée et retourne un résultat écrit si possible

DOMXPath mixed **evaluate** (string expression , DOMNode contextnode)

Exécute l'expression XPath *expression* et retourne un résultat écrit si possible.

8.28.87 DOMXPath->query()() : Évalue l'expression XPath donnée

DOMXPath DOMNodeList **query** (string *expression* , *DOMNode contextnode*)

Évalue l'expression *expression* XPath donnée

8.28.88 DOMXPath->registerNamespace()() : Enregistre l'espace de noms avec l'objet DOMXPath

DOMXPath bool **registerNamespace** (string *prefix* , string *namespaceURI*)

Enregistre l'espace de noms *namespaceURI* et le *prefix* avec l'objet DOMXPath.

8.28.89 dom_import_simplexml() : Transforme un objet DOMElement en un objet SimpleXMLElement

DOMElement **dom_import_simplexml** (SimpleXMLElement *node*)

dom_import_simplexml transforme l'objet SimpleXML *node* en un objet DOMElement . Le nouvel objet peut être utilisé en tant que noeud natif DOMElement *node*.

8.29 Fonctions DOM XML

8.29.1 Introduction

L'extension DOM XML a été reprise en main en PHP 4.3.0 pour être mieux compatible avec les standards DOM. L'extension contient encore de vieilles fonctions, mais elles ne sont plus utilisées. En particulier, les fonctions qui ne sont pas orientées objets ne doivent pas être utilisées.

Cette extension vous permet de générer des documents XML avec les API DOM. Elle fournit aussi une fonction `domxml_xmltree` qui transforme un fichier XML en tableau PHP. Actuellement, ce tableau est accessible uniquement en lecture. Cela ne signifie pas que vous ne pouvez pas le modifier, mais cela n'aurait aucun sens car `DomDocument_dump_mem` ne pourra pas prendre ces modifications en considération. Par conséquent, si vous voulez lire un fichier XML et écrire sa version modifiée, utilisez les fonctions `DomDocument_create_text_node`, `set_attribute`, etc. et finalement `DomDocument_dump_mem`.

Note
Cette extension a été déplacée dans le module PECL et ne sera plus intégrée dans PHP à partir de PHP 5.0.0.
Note
Cette extension n'est plus considérée comme expérimental. Néanmoins, elle ne sera jamais distribuée avec PHP 5 mais uniquement avec PHP 4. Si vous avez besoin du support DOM XML avec PHP 5, vous pouvez utiliser l'extension DOM . <code>domxml</code> n'est pas compatible avec l'extension DOM .

8.29.2 Pré-requis

Cette extension utilise la bibliothèque [GNOME XML](#). Téléchargez là, puis installez là. Vous aurez besoin de la version libxml-2.4.14 ou plus récent. Pour utiliser les fonctionnalités DOM XSLT vous pouvez utiliser la bibliothèque [libxslt](#) et les progrès des EXSLT (<http://www.exslt.org/>). Téléchargez et installez ces bibliothèques si vous envisagez d'exploiter ces fonctionnalités. Vous aurez besoin d'au moins libxslt-1.0.18.

8.29.3 Installation

Cette extension [PECL](#) n'est pas intégrée à PHP. Des informations comme les notes sur les nouvelles versions, les téléchargements, les sources des fichiers, les informations concernant les mainteneurs ainsi qu'un CHANGELOG, peuvent être trouvées ici : <http://pecl.php.net/package/domxml>.

En PHP 4, les sources de cette extension PECL peuvent être trouvées dans le dossier `ext/` avec les sources de PHP ou sur le lien PECL ci-dessous. Cette extension est uniquement disponible si PHP a été configuré lors de la compilation avec l'option `--with-dom=[DIR]`. Ajoutez `--with-dom-xslt=[DIR]` pour inclure le support DOM XSLT. DIR est le dossier d'installatoin de libxslt. Ajoutez `--with-dom-exslt=[DIR]` pour inclure le support DOM EXSLT. DIR est le dossier d'installation de libxsl.

Les utilisateurs de Windows doivent activer la bibliothèque `php_domxml.dll` dans le `php.ini` pour pouvoir utiliser ces fonctions. En PHP 4, cette bibliothèque DLL se trouve dans le dossier `extensions/` avec les binaires PHP pour Windows téléchargées. Vous pouvez télécharger ces

bibliothèques DLL pour ces extensions PECL depuis la page [de téléchargement de PHP](#) sous sur la page <http://snaps.php.net/> . En outre, il y a une DLL additionnelle qui doit être disponible à votre PATH système pour faire fonctionner cette extension. En PHP 4, elle se trouve dans le dossier dlls/ . Son nom : Pour PHP <= 4.2.0, c'est libxml2.dll . Pour PHP >= 4.3.0, c'est iconv.dll . Et depuis PHP 5.0.0, iconv est compilé dans les binaires PHP pour Windows par défaut, donc, aucune bibliothèque supplémentaire n'est nécessaire.

8.29.4 Fonctions dépréciées

Il y a un groupe de fonctions qui ne sont pas compatibles avec les standards DOM et qui ne devraient plus être utilisées. Ces fonctions sont listées dans la table ci-dessous. La fonction [DomNode_append_child](#) a vu son comportement changer. Désormais, elle ajoute un fils et non pas un frère. Si cela casse votre application, utilisez plutôt la fonction [DomNode_append_sibling](#) , qui ne fait pas partie des standards.

Ancienne fonction	Nouvelle fonction
xmlDoc	domxml_open_mem
xmlDocfile	domxml_open_file
domxml_new_xmlDoc	domxml_new_doc
domxml_dump_mem	DomDocument_dump_mem
domxml_dump_mem_file	DomDocument_dump_file
DomDocument_dump_mem_file	DomDocument_dump_file
DomDocument_add_root	DomDocument_create_element suivie de DomNode_append_child
DomDocument_dtd	DomDocument_doctype
DomDocument_root	DomDocument_document_element
DomDocument_children	DomNode_child_nodes
DomDocument_imported_node	Pas de remplacement.
DomNode_add_child	Créez un nouveau noeud avec DomDocument_create_element puis ajoutez-le avec DomNode_append_child .
DomNode_children	DomNode_child_nodes
DomNode_parent	DomNode_parent_node
DomNode_new_child	Créez un nouveau noeud avec DomDocument_create_element puis ajoutez-le avec DomNode_append_child .
DomNode_set_content	Créez un nouveau noeud avec DomDocument_create_text_node puis ajoutez-le avec DomNode_append_child .
DomNode_get_content	Le contenu est juste un noeud de texte, et est accessible via DomNode_child_nodes .
DomNode_set_content	Le contenu est juste un noeud de texte, et peut être ajouté avec DomNode_append_child .

8.29.5 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Constant	Value	Description
XML_ELEMENT_NODE (entier)	1	Node est un élément

XML_ATTRIBUTE_NODE (<u>entier</u>)	2	Node est un attribut
XML_TEXT_NODE (<u>entier</u>)	3	Node est un texte
XML_CDATA_SECTION_NODE (<u>entier</u>)	4	
XML_ENTITY_REF_NODE (<u>entier</u>)	5	
XML_ENTITY_NODE (<u>entier</u>)	6	Node est une entité comme
XML_PI_NODE (<u>entier</u>)	7	Node est une instruction de script
XML_COMMENT_NODE (<u>entier</u>)	8	Node est un commentaire
XML_DOCUMENT_NODE (<u>entier</u>)	9	Node est un document
XML_DOCUMENT_TYPE_NODE (<u>entier</u>)	10	
XML_DOCUMENT_FRAG_NODE (<u>entier</u>)	11	
XML_NOTATION_NODE (<u>entier</u>)	12	
XML_GLOBAL_NAMESPACE (<u>entier</u>)	1	
XML_LOCAL_NAMESPACE (<u>entier</u>)	2	
XML_HTML_DOCUMENT_NODE (<u>entier</u>)		
XML_DTD_NODE (<u>entier</u>)		
XML_ELEMENT_DECL_NODE (<u>entier</u>)		
XML_ATTRIBUTE_DECL_NODE (<u>entier</u>)		
XML_ENTITY_DECL_NODE (<u>entier</u>)		
XML_NAMESPACE_DECL_NODE (<u>entier</u>)		
XML_ATTRIBUTE_CDATA (<u>entier</u>)		
XML_ATTRIBUTE_ID (<u>entier</u>)		
XML_ATTRIBUTE_IDREF (<u>entier</u>)		
XML_ATTRIBUTE_IDREFS (<u>entier</u>)		
XML_ATTRIBUTE_ENTITY (<u>entier</u>)		
XML_ATTRIBUTE_NMTOKEN (<u>entier</u>)		
XML_ATTRIBUTE_NMTOKENS (<u>entier</u>)		
XML_ATTRIBUTE_ENUMERATION (<u>entier</u>)		
XML_ATTRIBUTE_NOTATION (<u>entier</u>)		
XPATH_UNDEFINED (<u>entier</u>)		
XPATH_NODESET (<u>entier</u>)		
XPATH_BOOLEAN (<u>entier</u>)		
XPATH_NUMBER (<u>entier</u>)		
XPATH_STRING (<u>entier</u>)		
XPATH_POINT (<u>entier</u>)		
XPATH_RANGE (<u>entier</u>)		
XPATH_LOCATIONSET (<u>entier</u>)		
XPATH_USERS (<u>entier</u>)		
XPATH_NUMBER (<u>entier</u>)		

8.29.6 Classes

L'API de ce module suit le standard DOM niveau 2 aussi fidèlement que possible. Par conséquent, l'API est totalement orientée objet. C'est une bonne idée d'avoir les standards DOM disponibles sous la main, lorsque vous utilisez ce module. Bien que cette API soit orientée objet, il existe de nombreuses fonctions qui peuvent être appelées d'une manière non objet, en passant l'objet à utiliser comme premier argument de la fonction. Ces fonctions sont essentiellement conservées

pour assurer la compatibilité avec les extensions plus anciennes, et ne doivent pas être utilisées.

Cette API diffère de l'API officielle DOM en deux points. Le premier est que tous les attributs des classes sont implémentés comme des fonctions du même nom. Le second est que les noms des fonctions suivent les standards PHP. Cela signifie que la fonction DOM lastChild() s'écrit en PHP last_child().

Ce module définit un grand nombre de classe, qui sont listés en incluant leur méthodes dans les tables suivantes. Les classes ayant un équivalent DOM sont nommées DOMxxx.

Nom de la classe	Classe parente
DomAttribute	DomNode
DomCDATA	DomNode
DomComment	DomCDATA : DomNode
DomDocument	DomNode
DomDocumentType	DomNode
DomElement	DomNode
DomEntity	DomNode
DomEntityReference	DomNode
DomProcessingInstruction	DomNode
DomText	DomCDATA : DomNode
Parser	Actuellement, toujours appelée DomParser
XPathContext	

Nom de la méthode	Nom de la fonction	Remarques
doctype	<u>DomDocument_doctype</u>	
document_element	<u>DomDocument_document_element</u>	
create_element	<u>DomDocument_create_element</u>	
create_text_node	<u>DomDocument_create_text_node</u>	
create_comment	<u>DomDocument_create_comment</u>	
create_cdata_section	<u>DomDocument_create_cdata_section</u>	
create_processing_instruction	<u>DomDocument_create_processing_instruction</u>	
create_attribute	<u>DomDocument_create_attribute</u>	
create_entity_reference	<u>DomDocument_create_entity_reference</u>	
get_elements_by_tagname	<u>DomDocument_get_elements_by_tagname</u>	
get_element_by_id	<u>DomDocument_get_element_by_id</u>	
dump_mem	<u>DomDocument_dump_mem</u>	Non standard DOM
dump_file	<u>DomDocument_dump_file</u>	Non standard DOM
html_dump_mem	<u>DomDocument_html_dump_mem</u>	Non standard DOM
xpath_init	xpath_init	Non standard DOM
xpath_new_context	xpath_new_context	Non standard DOM
xptr_new_context	xptr_new_context	Non standard DOM

Nom de la méthode	Nom de la fonction	Remarques
tagname	<u>DomElement_tagname</u>	
get_attribute	<u>DomElement_get_attribute</u>	
set_attribute	<u>DomElement_set_attribute</u>	

<code>remove_attribute</code>	<code>DomElement_remove_attribute</code>
<code>get_attribute_node</code>	<code>DomElement_get_attribute_node</code>
<code>get_elements_by_tagname</code>	<code>DomElement_get_elements_by_tagname</code>
<code>has_attribute</code>	<code>DomElement_has_attribute</code>

Nom de la méthode	Remarques
<code>DomNode_node_name</code>	
<code>DomNode_node_value</code>	
<code>DomNode_node_type</code>	
<code>DomNode_last_child</code>	
<code>DomNode_first_child</code>	
<code>DomNode_child_nodes</code>	
<code>DomNode_previous_sibling</code>	
<code>DomNode_next_sibling</code>	
<code>DomNode_parent_node</code>	
<code>DomNode_owner_document</code>	
<code>DomNode_insert_before</code>	
<code>DomNode_append_child</code>	
<code>DomNode_append_sibling</code>	Non standard DOM. Cette fonction émule le comportement précédent de <code>DomNode_append_child</code> .
<code>DomNode_remove_child</code>	
<code>DomNode_has_child_nodes</code>	
<code>DomNode_has_attributes</code>	
<code>DomNode_clone_node</code>	
<code>DomNode_attributes</code>	
<code>DomNode_unlink_node</code>	Non standard DOM
<code>DomNode_replace_node</code>	Non standard DOM
<code>DomNode_set_content</code>	Non standard DOM, déprécié
<code>DomNode_get_content</code>	Non standard DOM, déprécié
<code>DomNode_dump_node</code>	Non standard DOM
<code>DomNode_is_blank_node</code>	Non standard DOM

Nom de la méthode		Remarques
<code>name</code>	<code>DomAttribute_name</code>	
<code>value</code>	<code>DomAttribute_value</code>	
<code>specified</code>	<code>DomAttribute_specified</code>	

Nom de la méthode	Nom de la fonction	Remarques
<code>target</code>	<code>DomProcessingInstruction_target</code>	
<code>data</code>	<code>DomProcessingInstruction_data</code>	

Nom de la méthode	Nom de la fonction	Remarques
<code>add_chunk</code>	<code>Parser_add_chunk</code>	
<code>end</code>	<code>Parser_end</code>	

Nom de la méthode	Nom de la fonction	Remarques
-------------------	--------------------	-----------

eval	XPathContext_eval
eval_expression	XPathContext_eval_expression
register_ns	XPathContext_register_ns

Nom de la méthode	Nom de la fonction	Remarques
name	DomDocumentType_name	
entities	DomDocumentType_entities	
notations	DomDocumentType_notations	
public_id	DomDocumentType_public_id	
system_id	DomDocumentType_system_id	
internal_subset	DomDocumentType_internal_subset	

Les classes DomDtd sont dérivées de DomNode. DomComment est dérivée de DomCData.

8.29.7 Exemples

De nombreux exemples de cette partie requièrent une chaîne XML. Au lieu de répéter la chaîne dans tous les exemples, elle sera mise dans un fichier, qui sera inclus dans tous les exemples. Ce fichier inclus sera utilisé dans les exemples suivants dans cette section. Alternativement, vous pouvez créer un document XML et le lire avec la fonction [DomDocument_open_file](#).

Fichier d'inclusion exemple.inc pour la chaîne XML d'exemple
<pre> <?php \$xmlstr = " <!DOCTYPE chapter SYSTEM '/share/sgml/Norman_Walsh/db3xml110/db3xml110.dtd' [<!ENTITY sp \"spanish\">]> <chapter language='en'><title language='en'>Title</title> <para language='ge'> &sp; <informaltable ID='findme' language='&sp;'> <tgroup cols='3'> <tbody> <row><entry>a1</entry><entry morerows='1'>b1</entry><entry>c1</entry></row> <row><entry>a2</entry><entry>c2</entry></row> <row><entry>a3</entry><entry>b3</entry><entry>c3</entry></row> </tbody> </tgroup> </informaltable> </para> </chapter>"; ?> </pre>

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Fonctions dépréciées](#)
- [Constantes pré-définies](#)
- [Classes](#)
- [Exemples](#)
- [DomAttribute->name](#)
- [DomAttribute->set_value](#)

- [DomAttribute->specified](#)
- [DomAttribute->value](#)
- [DomDocument->add_root](#)
- [DomDocument->create_attribute](#)
- [DomDocument->create_cdata_section](#)
- [DomDocument->create_comment](#)
- [DomDocument->create_element_ns](#)
- [DomDocument->create_element](#)
- [DomDocument->create_entity_reference](#)
- [DomDocument->create_processing_instruction](#)
- [DomDocument->create_text_node](#)
- [DomDocument->doctype](#)
- [DomDocument->document_element](#)
- [DomDocument->dump_file](#)
- [DomDocument->dump_mem](#)
- [DomDocument->get_element_by_id](#)
- [DomDocument->get_elements_by_tagname](#)
- [DomDocument->html_dump_mem](#)
- [DomDocument->xinclude](#)
- [DomDocumentType->entities\(\)](#)
- [DomDocumentType->internal_subset\(\)](#)
- [DomDocumentType->name\(\)](#)
- [DomDocumentType->notations\(\)](#)
- [DomDocumentType->public_id\(\)](#)
- [DomDocumentType->system_id\(\)](#)
- [DomElement->get_attribute_node\(\)](#)
- [DomElement->get_attribute\(\)](#)
- [DomElement->get_elements_by_tagname\(\)](#)
- [DomElement->has_attribute\(\)](#)
- [DomElement->remove_attribute\(\)](#)
- [DomElement->set_attribute\(\)](#)
- [DomElement->tagname\(\)](#)
- [DomNode->add_namespace](#)
- [DomNode->append_child](#)
- [DomNode->append_sibling](#)
- [DomNode->attributes](#)
- [DomNode->child_nodes](#)
- [DomNode->clone_node](#)
- [DomNode->dump_node](#)
- [DomNode->first_child](#)
- [DomNode->get_content](#)
- [DomNode->has_attributes](#)
- [DomNode->has_child_nodes](#)
- [DomNode->insert_before](#)
- [DomNode->is_blank_node](#)
- [DomNode->last_child](#)
- [DomNode->next_sibling](#)
- [DomNode->node_name](#)
- [DomNode->node_type](#)
- [DomNode->node_value](#)
- [DomNode->owner_document](#)
- [DomNode->parent_node](#)
- [DomNode->prefix](#)
- [DomNode->previous_sibling](#)

- [DOMNode->remove_child](#)
- [DOMNode->replace_child](#)
- [DOMNode->replace_node](#)
- [DOMNode->set_content](#)
- [DOMNode->set_name](#)
- [DOMNode->set_namespace](#)
- [DOMNode->unlink_node](#)
- [DomProcessingInstruction->data](#)
- [DomProcessingInstruction->target](#)
- [domxml_new_doc](#)
- [domxml_open_file](#)
- [domxml_open_mem](#)
- [domxml_version](#)
- [domxml_xmltree](#)
- [domxml_xslt_stylesheet_doc](#)
- [domxml_xslt_stylesheet_file](#)
- [domxml_xslt_stylesheet](#)
- [domxml_xslt_version](#)
- [DomXsltStylesheet->process\(\)](#)
- [DomXsltStylesheet->result_dump_file\(\)](#)
- [DomXsltStylesheet->result_dump_mem\(\)](#)
- [xpath_eval_expression](#)
- [xpath_eval](#)
- [xpath_new_context](#)
- [xpath_register_ns_auto](#)
- [xpath_register_ns](#)
- [xptr_eval](#)
- [xptr_new_context](#)

8.29.9 DomAttribute->set_value() : Spécifie une valeur à un attribut

DomAttribute bool **set_value** (string content)

Cette fonction spécifie une valeur à un attribut.

8.29.10 DomAttribute->specified() : Vérifie si un attribut est spécifié

DomAttribute bool **specified** (void)

Vérifie si une valeur d'attribut a été explicitement donnée dans le document original.

Note

Cette méthode n'est pas encore implémentée.

8.29.11 DomAttribute->value() : Retourne la valeur d'un attribut

DomAttribute string **value** (void)

DomAttribute->value retourne la valeur de l'attribut.

8.29.12 DomDocument->add_root() : Ajoute un noeud

domelement **DomDocument->add_root** (string name)

DomDocument->add_root ajoute un noeud racine au document dom, et retourne le nouveau noeud. Le nom de l'élément créé est fourni par le paramètre name .

Création d'un en-tête de document HTML

```
<?php
$doc = domxml_new_doc("1.0");
$root = $doc->add_root("html");
$head = $root->new_child("head", "");
$head->new_child("title", "Voilà le titre");
echo htmlentities($doc->dump_mem());
?>
```

8.29.13 DomDocument->create_attribute() : Crée un nouvel attribut

domattribute **DomDocument->create_attribute** (string name , string value)

DomDocument->create_attribute retourne une nouvelle instance de la classe DomAttribute . Le nom de l'attribut sera name , et sa valeur sera value . Ce noeud ne sera pas utilisé dans le document jusqu'à ce qu'il soit inséré dans le document avec la fonction domnode_append_child .

DomDocument->create_attribute retourne FALSE si une erreur survient.

Voir aussi domnode_append_child , domdocument_create_element , domdocument_create_text , domdocument_create_cdata_section , domdocument_create_processing_instruction , domdocument_create_entity_reference et domnode_insert_before .

8.29.14 DomDocument->create_cdata_section() : Crée un nouveau noeud CDATA

domcdata **DomDocument->create_cdata_section** (string content)

DomDocument->create_cdata_section retourne une nouvelle instance de la classe DomCData . Le contenu du CDATA sera content . Le noeud ne sera pas affiché dans le document, à moins qu'il ne soit inséré avec domnode_append_child .

DomDocument->create_cdata_section retourne FALSE si une erreur survient.

Voir aussi domnode_append_child , domdocument_create_element , domdocument_create_text , domdocument_create_attribute , domdocument_create_processing_instruction , domdocument_create_entity_reference et domnode_insert_before .

8.29.15 DomDocument->create_comment() : Crée un nouveau noeud de commentaire

domcomment **DomDocument->create_comment** (string content)

DomDocument->create_comment retourne une nouvelle instance de la classe DomComment . Le contenu du commentaire sera content . Le noeud ne sera pas affiché dans le document, à moins qu'il ne soit inséré avec domnode_append_child .

DomDocument->create_comment retourne FALSE si une erreur survient.

Voir aussi domnode_append_child , domdocument_create_element , domdocument_create_text , domdocument_create_attribute , domdocument_create_processing_instruction , domdocument_create_entity_reference et domnode_insert_before .

8.29.16 DomDocument->create_element_ns() : Crée un nouvel élément avec un espace de noms associé

domelement **DomDocument->create_element_ns** (string uri , string name , string prefix)

DomDocument->create_element_ns retourne une nouvelle instance de la classe DomElement . Le nom de balise de l'élément sera content . L'URI de l'espace de noms est la valeur passé dans le paramètre uri . S'il y a déjà un espace de nom déclaré avec la même URI dans le document, le préfixe de ce dernier sera réservé, et sinon, il faudra en fournir un avec le paramètre prefix ou bien en générer un. Le noeud ne sera pas affiché dans le document, à moins qu'il ne soit inséré avec domnode_append_child .

DomDocument->create_element_ns retourne FALSE si une erreur survient.

Voir aussi domdocument_create_element_ns , domnode_add_namespace , domnode_set_namespace , domnode_append_child , domdocument_create_text , domdocument_create_comment , domdocument_create_attribute , domdocument_create_processing_instruction , domdocument_create_entity_reference et domnode_insert_before .

8.29.17 DomDocument->create_element() : Crée un nouveau noeud élément

domelement **DomDocument->create_element** (string name)

DomDocument->create_element retourne une nouvelle instance de la classe DomElement . Le nom de balise l'élément sera name . Le noeud ne sera pas affiché dans le document, à moins qu'il ne soit inséré avec domnode_append_child .

DomDocument->create_processing_instruction retourne FALSE si une erreur survient.

Voir aussi domdocument_create_element_ns , domnode_append_child , domdocument_create_text , domdocument_create_comment , domdocument_create_attribute , domdocument_create_processing_instruction , domdocument_create_entity_reference et domnode_insert_before .

8.29.18 DomDocument->create_entity_reference() : Crée une entité référence

dumentityreference **DomDocument->create_entity_reference** (string content)

DomDocument->create_entity_reference retourne une nouvelle instance de la classe DomEntityReference . Le contenu de la référence d'entité est passé dans le paramètre content . Le noeud ne sera pas affiché dans le document, à moins qu'il ne soit inséré avec domnode_append_child .

DomDocument->create_entity_reference retourne FALSE si une erreur survient.

Voir aussi domnode_append_child , domdocument_create_element , domdocument_create_text , domdocument_create_cdata_section , domdocument_create_processing_instruction , domdocument_create_attribute et domnode_insert_before .

8.29.19 DomDocument->create_processing_instruction() : Crée un nouveau noeud de traitement d'insctructions (PI)

domprocessinginstruction **DomDocument->create_processing_instruction** (string content)

DomDocument->create_processing_instruction retourne une nouvelle instance de la classe DomCData . Le contenu de l'unité de traitement sera content . Le noeud ne sera pas affiché dans le document, à moins qu'il ne soit inséré avec domnode_append_child .

DomDocument->create_processing_instruction retourne FALSE si une erreur survient.

Voir aussi domnode_append_child , domdocument_create_element , domdocument_create_text , domdocument_create_cdata_section , domdocument_create_attribute , domdocument_create_entity_reference et domnode_insert_before .

8.29.20 DomDocument->create_text_node() : Crée un nouveau noeud de texte

domtext **DomDocument->create_text_node** (string content)

DomDocument->create_text_node retourne une nouvelle instance de la classe DomText . Le contenu de ce noeud est content . Le noeud ne sera pas affiché dans le document, à moins qu'il ne soit inséré avec domnode_append_child .

La fonction retourne FALSE en cas d'erreur.

Voir aussi domnode_append_child , domdocument_create_element , domdocument_create_comment , domdocument_create_text , domdocument_create_attribute , domdocument_create_processing_instruction , domdocument_create_entity_reference et domnode_insert_before .

8.29.21 DomDocument->doctype() : Retourne le type de document

domdocumenttype **DomDocument->doctype** (void)

DomDocument->doctype retourne un objet de la classe DomDocumenttype . Dans les versions de PHP antérieures à la version 4.3, c'était la classe Dtd , mais les standards DOM ne disposent pas d'une telle classe.

Voir aussi les méthodes de la classe DomDocumentType .

8.29.22 DomDocument->document_element() : Retourne le noeud racine d'un document DOM XML

domelement **DomDocument->document_element** (void)

DomDocument->document_element retourne le noeud racine d'un document DOM XML.

L'exemple suivant retourne simplement l'élément nommé CHAPTER , puis l'affiche. L'autre noeud (le commentaire), n'est pas retourné.

Récupération du noeud racine avec DomDocument->document_element

```
<?php
include("example.inc");

if (!$dom = domxml_open_mem($xmlstr)) {
    echo "Erreur lors de l'analyse du document\n";
    exit;
}

$root = $dom->document_element();
print_r($root);
?>
```

L'exemple ci-dessus va afficher :

```
domelement Object
(
    [type] => 1
    [tagname] => chapter
    [0] => 6
    [1] => 137960648
)
```

8.29.23 DomDocument->dump_file() : Convertit un document DOM XML en fichier

string **DomDocument->dump_file** (string filename , bool compressionmode , bool format)

DomDocument->dump_file crée un document XML à partir de sa représentation interne DOM. DomDocument->dump_mem est généralement appelée après la construction d'un document à partir de zéro, comme dans l'exemple ci-dessous. Le paramètre format spécifie si l'affichage doit être formaté proprement ou pas. filename est le nom du fichier dans lequel le document XML sera sauvegardé, et compressionmode indique si le fichier doit être compressé ou pas.

Création d'un en-tête de document HTML avec DomDocument->dump_file

```
<?php
$doc = domxml_new_doc("1.0");
$root = $doc->create_element("HTML");
$root = $doc->append_child($root);
$head = $doc->create_element("HEAD");
$head = $root->append_child($head);
```

```

$title = $doc->create_element("TITLE");
$title = $head->append_child($title);
$text = $doc->create_text_node("This is the title");
$text = $title->append_child($text);
$doc->dump_file("/tmp/test.xml", false, true);
?>

```

Voir aussi [domdocument_dump_mem](#) et [domdocument_html_dump_mem](#) .

8.29.24 DomDocument->dump_mem() : Convertit l'arbre XML interne en une chaîne de caractères

string **DomDocument->dump_mem** (*bool format* , *string encoding*)

[DomDocument->dump_mem](#) crée un document XML à partir de sa représentation interne DOM. [DomDocument->dump_mem](#) est généralement appelée après la construction d'un document à partir de zéro, comme dans l'exemple ci-dessous. Le paramètre format spécifie si l'affichage doit être formaté proprement ou pas.

Création d'un en-tête de document HTML avec [DomDocument->dump_mem](#)

```

<?php
$doc = domxml_new_doc("1.0");
$root = $doc->create_element("HTML");
$root = $doc->append_child($root);
$head = $doc->create_element("HEAD");
$head = $root->append_child($head);
$title = $doc->create_element("TITLE");
$title = $head->append_child($title);
$text = $doc->create_text_node("This is the title");
$text = $title->append_child($text);
echo "<pre>";
echo htmlentities($doc->dump_mem(true));
echo "</pre>";
?>

```

Note

Le premier paramètre a été ajouté en PHP 4.3.0.

Voir aussi [domdocument_dump_file](#) et [domdocument_html_dump_mem](#) .

8.29.25 DomDocument->get_element_by_id() : Recherche un élément avec son identifiant

domelement **DomDocument->get_element_by_id** (string id)

[DomDocument->get_element_by_id](#) est similaire à [domdocument_get_elements_by_tagname](#) mais recherche un élément grâce à l'identifiant passé en paramètre id . Suivant les standards DOM, cela impose l'utilisation d'une DTD qui définit l'attribut ID du type ID, même si l'implémentation courante fait une simple recherche xpath sur `//*[@ID = '%s']` . Ce n'est pas conforme aux standard DOM, qui demandent le retour de la valeur NULL s'il n'est pas possible de savoir quel attribut est de type id. Ce comportement sera probablement corrigé, alors ne vous fiez pas totalement à ce comportement actuel.

Voir aussi [domdocument_get_elements_by_tagname](#) .

8.29.26 DomDocument->get_elements_by_tagname() : Retourne un tableau avec noeuds pour le nom de tag donné dans le document ou un tableau vide si non trouvé

array **DomDocument->get_elements_by_tagname** (string name)

Voir aussi [domdocument_add_root](#) .

8.29.27 DomDocument->html_dump_mem() : Convertit l'arbre XML interne en une chaîne de caractères au format HTML

string **DomDocument->html_dump_mem** (void)

[DomDocument->html_dump_mem](#) crée un document HTML à partir de sa représentation interne DOM. [DomDocument->dump_mem](#) est généralement appelée après la construction d'un document à partir de zéro, comme dans l'exemple ci-dessous.

Création d'un en-tête de document HTML avec DomDocument->html_dump_mem
<pre> <?php // Création du document \$doc = domxml_new_doc("1.0"); \$root = \$doc->create_element("html"); \$root = \$doc->append_child(\$root); \$head = \$doc->create_element("head"); \$head = \$root->append_child(\$head); \$title = \$doc->create_element("title"); \$title = \$head->append_child(\$title); \$text = \$doc->create_text_node("Ceci est le titre"); \$text = \$title->append_child(\$text); echo \$doc->html_dump_mem(); ?> </pre> <p>L'exemple ci-dessus va afficher :</p> <pre> <html><head><title>Ceci est le titre</title></head></html> </pre>

Voir aussi [domdocument_dump_file](#) et [domdocument_html_dump_mem](#) .

8.29.28 DomDocument->xinclude() : Remplace XIncludes dans un objet DomDocument

int **DomDocument->xinclude** (void)

[DomDocument->xinclude](#) substitue [XIncludes](#) dans un objet DomDocument.

Substitution des Xincludes

```

<?php
// include.xml contient :
// <child>test</child>

$xml = '
<root xmlns:xi="http://www.w3.org/2001/XInclude">
  <xi:include href="include.xml">
    <xi:fallback>
      <error>xinclude: include.xml n\'a pas été trouvé</error>
    </xi:fallback>
  </xi:include>
</root>';

$domxml = domxml_open_mem($xml);
$domxml->xinclude();

echo $domxml->dump_mem();

?>

```

L'exemple ci-dessus va afficher :

```

<root xmlns:xi="http://www.w3.org/2001/XInclude">
<child>test</child>
</root>

```

Si include.xml n'existe pas, vous verrez :

```

<root xmlns:xi="http://www.w3.org/2001/XInclude">
<error>xinclude:dom.xml n'a pas été trouvé</error>
</root>

```

8.29.29 DomDocumentType->entities()() : Retourne la liste des entités

DomDocumentType array **entities** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.29.30 DomDocumentType->internal_subset()() : Retourne le sous ensemble interne (internal subset)

DomDocumentType bool **internal_subset** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.29.31 DomDocumentType->name()() : Retourne le nom du type de document

DomDocumentType string **name** (void)

Cette fonction retourne le nom du type de document.

8.29.32 DomDocumentType->notations()() : Retourne la liste des notations

DomDocumentType array **notations** (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.29.33 DomDocumentType->public_id()() : Retourne l'identifiant public du type de document

DomDocumentType string **public_id** (void)

Cette fonction retourne l'identifiant public du type de document.

8.29.34 DomDocumentType->system_id()() : Retourne l'identifiant système du type de document

DomDocumentType string **system_id** (void)

Retourne l'identifiant système du type de document.

8.29.35 DomElement->get_attribute_node()() : Retourne la valeur d'un attribut

DomElement DomAttribute **get_attribute_node** (string name)

Retourne le noeud de l'attribut donné dans l'élément courant.

8.29.36 DomElement->get_attribute()() : Retourne la valeur d'un attribut

DomElement string **get_attribute** (string name)

Retourne la valeur de l'attribut donné dans l'élément courant.

Depuis PHP 4.3, si aucun attribut de nom name n'est trouvé, une chaîne de caractères vide est retournée.

8.29.37 DomElement->get_elements_by_tagname()() : Lit les éléments par nom de balise

DomElement array **get_elements_by_tagname** (string name)

Récupère tous les sous éléments avec un nom spécifique name à l'intérieur de l'élément courant.

8.29.38 DomElement->has_attribute()() : Vérifie si un attribut existe

DomElement bool **has_attribute** (string name)

Cette fonction vérifie l'existence d'un attribut nommé name dans le noeud courant.

8.29.39 DomElement->remove_attribute()() : Supprime un attribut

DomElement bool **remove_attribute** (string name)

Supprime un attribut du noeud courant DomElement .

8.29.40 DomElement->set_attribute()() : Définit la valeur d'un attribut

DomElement DomAttribute **set_attribute** (string name , string value)

Définit l'attribut de nom name , avec la valeur value .

8.29.41 DomElement->>tagname()() : Retourne le nom de l'élément courant

DomElement string **tagname** (void)

Retourne le nom du noeud courant. Appeler cette fonction revient à accéder à la propriété tagname ou appeler DOMNode->node_name sur le noeud courant.

8.29.42 DomNode->add_namespace() : Ajoute une déclaration d'espace de nom à un noeud

DOMNode bool **add_namespace** (string uri , string prefix)

Cette méthode ajoute une déclaration de nom à un noeud.

Note
Cette méthode ne fait pas parti des spécifications DOM.

8.29.43 DomNode->append_child() : Ajoute un nouveau fils à la fin des enfants

DOMNode DOMNode **append_child** (DOMNode newnode)

Cette fonction ajoute un fils à une liste existante de fils ou crée une nouvelle liste de fils.

8.29.44 DomNode->append_sibling() : Ajoute un frère à un noeud

domelement **DOMNode->append_sibling** (domelement newnode)

DOMNode->append_sibling ajoute le noeud newnode aux fils du noeud courant. newnode peut être créé avec les fonctions domdocument_create_element , domdocument_create_text , etc. ou simplement en utilisant un autre noeud.

Avant d'être ajouté, un noeud est d'abord dupliqué. Par conséquent, le nouveau fils est une copie qui peut être modifiée sans altérer le noeud utilisé dans cette fonction. Si le noeud passé a des fils, ils seront aussi dupliqués, ce qui rend cette fonction pratique pour copier des documents XML complexes. La valeur retournée est le noeud fils ajouté. Si vous voulez apporter des modifications au noeud fils, vous devez utiliser le noeud retourné.

DOMNode->append_sibling a été ajoutée pour fournir un remplaçant à domnode_append_child , tel qu'il fonctionnait en PHP 4.2.

Voir aussi domnode_append_before .

8.29.45 DomNode->attributes() : Retourne la liste des attributs

array **DOMNode->attributes** (void)

DOMNode->attributes retourne un tableau avec les attributs du noeud si le noeud est de type XML_ELEMENT_NODE .

(PHP >= 4.3 uniquement) Si aucun attribut n'est trouvé, NULL est retourné.

8.29.46 DomNode->child_nodes() : Retourne les fils d'un noeud

array **DOMNode->child_nodes** (void)

DOMNode->child_nodes retourne tous les fils d'un noeud.

Voir aussi domnode_next_sibling et domnode_previous_sibling .

8.29.47 DomNode->clone_node() : Clone un noeud

domelement **DOMNode->clone_node** (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.29.48 DomNode->dump_node() : Convertit un noeud en chaîne

string **DOMNode->dump_node** (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Voir aussi domdocument_dump_mem .

8.29.49 DomNode->first_child() : Retourne le premier fils du noeud

domelement **DomNode->first_child** (void)

DomNode->first_child retourne le premier fils du noeud.

(PHP >= 4.3 uniquement) Si aucun noeud fils n'est trouvé, NULL est retourné.

Voir aussi domnode_last_child , domnode_next_sibling et domnode_previous_sibling .

8.29.50 DomNode->get_content() : Retourne le contenu du noeud

string **DomNode->get_content** (void)

Cette fonction retourne le contenu du noeud courant.

Obtenir un contenu

```
<?php
if (!$dom = domxml_open_mem($xmlstr)) {
    echo "Erreur pendant le passage du document\n";
    exit;
}

$root = $dom->document_element();

$node_array = $root->get_elements_by_tagname("element");

for ($i = 0; $i<count($node_array); $i++) {
    $node = $node_array[$i];
    echo "L'élément[$i] est : " . $node->get_content();
}

?>
```

8.29.51 DomNode->has_attributes() : Vérifie si un noeud a des attributs

bool **DomNode->has_attributes** (void)

DomNode->has_attributes vérifie si un noeud a des attributs.

Voir aussi domnode_has_child_nodes .

8.29.52 DomNode->has_child_nodes() : Vérifie si un noeud a des fils

bool **DomNode->has_child_nodes** (void)

DomNode->has_child_nodes vérifie si un noeud a des fils.

Voir aussi domnode_child_nodes .

8.29.53 DomNode->insert_before() : Insère un nouveau noeud fils

domelement **DomNode->insert_before** (domelement newnode , domelement refnode)

DomNode->insert_before insère le nouveau noeud fils newnode juste avant le noeud fils refnode . La valeur retournée est la valeur du noeud inséré. Si vous devez faire des modifications sur le fils, vous devez utiliser le noeud retourné.

(PHP >= 4.3 uniquement) Si newnode fait déjà partie du document, il sera d'abord déconnecté de son contexte courant. Si refnode vaut NULL alors newnode sera inséré à la fin de la liste des fils.

domnode_insert_before est très similaire à domnode_append_child comme l'exemple ci-dessous le montre : il réalise la même fonction que l'exemple de domnode_append_child .

Ajouter un noeud fils DOM XML

```
include("example.inc");

if(!$dom = domxml_open_mem($xmlstr)) {
    echo "Erreur durant l'analyse du document\n";
    exit;
}

$elements = $dom->get_elements_by_tagname("informaltable");
print_r($elements);
$element = $elements[0];

$newnode = $element->insert_before($element, $element);
$children = $newnode->children();
$attr = $children[1]->set_attribute("align", "left");

echo "<pre>";
$xmlfile = $dom->dump_mem();
echo htmlentities($xmlfile);
echo "</pre>";
```

Voir aussi domnode_append_child .

8.29.54 DomNode->is_blank_node() : Vérifie si un noeud est blanc

bool **DomNode->is_blank_node** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.29.55 DomNode->last_child() : Retourne le dernier fils du noeud

domelement **DomNode->last_child** (void)

DomNode->last_child retourne le dernier fils du noeud.

(PHP >= 4.3 uniquement) Si aucun fils n'est trouvé, NULL est retourné.

Voir aussi domnode_first_child , domnode_next_sibling et domnode_previous_sibling .

8.29.56 DomNode->next_sibling() : Retourne le noeud frère suivant

domelement **DomNode->next_sibling** (void)

DomNode->next_sibling retourne le prochain noeud fils du noeud courant. S'il n'y a pas de noeud fils suivant, cette fonction retournera FALSE (< PHP 4.3) ou NULL (>= 4.3). Vous pouvez utiliser cette fonction pour passer en revue tous les fils d'un noeud tel que présenté dans cet exemple.

Listage des noeuds fils
<pre><?php include("exemple.inc"); if(!\$dom = domxml_open_mem(\$xmlstr)) { echo "Erreur d'analyse du document\n"; exit; } \$elements = \$dom->get_elements_by_tagname("tbody"); \$element = \$elements[0]; \$child = \$element->first_child(); while(\$child) { print_r(\$child); \$child = \$child->next_sibling(); } ?></pre>

Voir aussi [domnode_previous_sibling](#) .

8.29.57 DomNode->node_name() : Retourne le nom du noeud

string **DomNode->node_name** (void)

DomNode->node_name retourne le nom du noeud. Le nom a différentes significations suivant le type de noeud, tel que présenté ci-dessous :

Type	Signification
DomAttribute	Valeur de l'attribut
DomAttribute	
DomCDATASection	#cdata-section
DomComment	#comment
DomDocument	#document
DomDocumentType	Nom du type de document
DomElement	Nom de la balise
DomEntity	Nom de l'entité
DomEntityReference	Nom de la référence de l'entité
DomNotation	Nom de la notation
DomProcessingInstruction	target
DomText	#text

8.29.58 DomNode->node_type() : Retourne le type de noeud

int **DomNode->node_type** (void)

DomNode->node_type retourne le type de noeud. Tous les types possibles sont listés dans l'introduction.

```

<?php
include 'example.inc';

$dom = domxml_open_mem($xmlstr);

$chapter = $dom->document_element();

// Regardons les éléments contenus dans "chapter"
foreach($chapter->child_nodes() as $node) {
    if ($node->node_type() == XML_ELEMENT_NODE) {
        echo $node->node_name() . "\n";
    }
}
?>

```

L'exemple ci-dessus va afficher :

title
para

8.29.59 DomNode->node_value() : Retourne la valeur d'un noeud

string **DomNode->node_value** (void)

DomNode->node_value retourne la valeur d'un noeud. La valeur a différente signification suivant le type de noeud, tel que présenté ci-dessous :

Type	Signification
DomAttribute	Valeur de l'attribut
DomAttribute	
DomCDataSection	Contenu
DomComment	Contenu du commentaire
DomDocument	NULL
DomDocumentType	NULL
DomElement	NULL
DomEntity	NULL
DomEntityReference	NULL
DomNotation	NULL
DomProcessingInstruction	Tout le document sans la cible (target)
DomText	Contenu du texte

8.29.60 DomNode->owner_document() : Retourne le document auquel appartient ce noeud DOM XML

domdocument **DomNode->owner_document** (void)

DomNode->owner_document retourne le document auquel appartient ce noeud.

L'exemple suivant affiche deux listes identiques de fils.

```
Liste les noeuds sous DOM XML

<?php
$doc = domxml_new_doc("1.0");
$node = $doc->create_element("para");
$node = $doc->append_child($node);
$children = $doc->children();
print_r($children);

$doc2 = $node->owner_document();
$children = $doc2->children();
print_r($children);
?>
```

Voir aussi domnode_insert_before .

8.29.61 DomNode->parent_node() : Retourne le père d'un noeud DOM XML

domnode **DomNode->parent_node** (void)

DomNode->parent_node retourne le père d'un noeud.

(PHP >= 4.3 uniquement) Si aucun parent n'est trouvé, NULL est retourné.

L'exemple ci-dessous affiche deux listes identiques de noeuds fils

```
Trouver le document d'un noeud avec DOM XML

<?php
$doc = domxml_new_doc("1.0");
$node = $doc->create_element("para");
$node = $doc->append_child($node);
$children = $doc->children();
print_r($children);

$doc2 = $node->parent_node();
$children = $doc2->children();
print_r($children);
?>
```

8.29.62 DomNode->prefix() : Retourne le préfixe d'espace de nom d'un noeud

string **DomNode->prefix** (void)

DomNode->prefix retourne le préfixe d'espace de nom d'un noeud.

8.29.63 DomNode->previous_sibling() : Retourne le frère précédent d'un noeud

domelement **DomNode->previous_sibling** (void)

DomNode->previous_sibling retourne le frère précédent d'un noeud. S'il n'y a pas de noeud précédent, DomNode->previous_sibling retournera FALSE (< PHP 4.3) ou NULL (>= PHP 4.3). Vous pouvez utiliser cette fonction pour lister tous les fils d'un noeud.

Voir aussi domnode_next_sibling .

8.29.64 DomNode->remove_child() : Supprime un fils de la liste des noeuds fils

domtext **DomNode->remove_child** (domtext oldchild)

DomNode->remove_child supprime le fils oldchild de la liste des noeuds fils du noeud courant. Si le fils n'a pu être retiré, ou si ce n'est pas un fils du noeud courant, DomNode->remove_child retournera FALSE . Si le fils a pu être retiré, DomNode->remove_child le retournera.

Supprimer un noeud en DOM XML

```
<?php
include("exemple.inc");

if(!$dom = domxml_open_mem($xmlstr)) {
    echo "Erreur lors de l'analyse d'un document\n";
    exit;
}

$elements = $dom->get_elements_by_tagname("tbody");
$element = $elements[0];
$children = $element->child_nodes();
$child = $element->remove_child($children[0]);

echo "<pre>";
$xmlfile = $dom->dump_mem(true);
echo htmlentities($xmlfile);
echo "</pre>";
?>
```

Voir aussi domnode_append_child .

8.29.65 DomNode->replace_child() : Remplace un noeud fils

domelement **DomNode->replace_child** (domelement oldnode , domelement newnode)

(PHP 4.2) DomNode->replace_child remplace le noeud fils oldnode avec le nouveau noeud newnode . Si le nouveau noeud est déjà un fils, il ne sera pas ajouté une seconde fois. Si l'ancien noeud ne peut être trouvé, DomNode->replace_child retournera FALSE . Si le remplacement a pu avoir lieu, oldnode est retourné.

(PHP 4.3) DomNode->replace_child remplace le noeud fils oldnode avec le nouveau noeud newnode , même si newnode est déjà fils. Si le nouveau noeud est déjà inséré dans le document, il sera retiré de son contexte courant.. Si DomNode->replace_child ne peut trouver l'ancien noeud,

elle retournera FALSE . Si le remplacement a pu avoir lieu, oldnode est retourné.

Voir aussi [domnode_append_child](#) .

8.29.66 DomNode->replace_node() : Remplace un noeud DomXML

domelement **DomNode->replace_node** (domelement newnode)

(PHP 4.2) [DomNode->replace_node](#) remplace un noeud par newnode . Avant le remplacement, newnode est copié s'il a un parent, pour s'assure qu'un noeud qui est déjà dans le document n'est pas inséré une nouvelle fois. Ce comportement impose de faire toutes les modifications sur le noeud avant le remplacement, ou de le relire après le remplacement, avec des fonctions comme [domnode_first_child](#) , [domnode_child_nodes](#) , etc.

(PHP 4.3) [DomNode->replace_node](#) remplace un noeud par le noeud newnode . Il n'est plus copié du tout. Si newnode était déjà inséré dans le document, il sera d'abord désolidarisé de son contexte courant. Si le remplacement réussi, le noeud est retourné.

Voir aussi [domnode_append_child](#) .

8.29.67 DomNode->set_content() : Modifie le contenu d'un noeud

bool **DomNode->set_content** (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.29.68 DomNode->set_name() : Modifie le nom d'un noeud

bool **DomNode->set_name** (void)

[DomNode->set_name](#) modifie le nom d'un noeud.

Voir aussi [domnode_node_name](#) .

8.29.69 DomNode->set_namespace() : Modifie l'espace de noms d'un noeud

void **DomNode->set_namespace** (string uri , string prefix)

DomNode->set_namespace remplace l'espace de noms courant du noeud par uri . S'il y a une déclaration d'espace de nom de même URI dans l'un des noeuds parents, le préfixe de ce parent sera alors utilisé. Sinon, la fonction prendre le préfixe fourni dans le paramètre optionnel prefix ou bien en générera un aléatoirement.

Voir aussi [domdocument_create_element_ns](#) et [domnode_add_namespace](#) .

8.29.70 DomNode->unlink_node() : Efface un noeud

void **DomNode->unlink_node** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.29.71 DomProcessingInstruction->data() : Retourne les données du noeud l'instruction en cours

DomProcessingInstruction string **data** (void)

Cette méthode retourne les données du noeud de l'instruction en cours.

8.29.72 DomProcessingInstruction->target() : Retourne la cible du noeud de l'instruction en cours

DomProcessingInstruction string **target** (void)

Cette méthode retourne la cible de l'instruction en cours.

8.29.73 domxml_new_doc() : Crée un document XML vide

DomDocument **domxml_new_doc** (string version)

Crée un nouveau document Dom à partir de zéro et le retourne.

8.29.74 domxml_open_file() : Crée un objet DOM à partir d'un fichier XML

DomDocument **domxml_open_file** (string filename , int mode , array error)

Cette fonction analyse le document XML dans le fichier donné.

8.29.75 domxml_open_mem() : Crée un objet DOM pour un document XML

DomDocument **domxml_open_mem** (string str , int mode , array error)

Cette fonction analyse le document XML dans le fichier donné.

8.29.76 domxml_version() : Lit le numéro de version de la bibliothèque XML

string **domxml_version** (void)

Lit le numéro de version de la bibliothèque XML actuellement utilisé.

8.29.77 domxml_xmltree() : Crée un arbre d'objets PHP à partir d'un document XML

DomDocument **domxml_xmltree** (string str)

Cette fonction analyse le document XML passé dans la chaîne str et retourne un arbre d'objets PHP, représentant le document.

Cette fonction est isolée des autres, ce qui signifie que vous ne pouvez pas accéder aux objets de l'arbre avec les autres fonctions. Modifier l'arbre, par exemple en ajoutant des noeuds, n'a pas de sens car il n'y a pas moyen de convertir cet arbre fichier XML.

Cependant, cette fonction est pratique si vous voulez lire un fichier XML et en étudier le contenu.

8.29.78 domxml_xslt_stylesheet_doc() : Crée un objet DomXsltStylesheet à partir d'un objet DomDocument

DomXsltStylesheet **domxml_xslt_stylesheet_doc** (DomDocument xsl_doc)

Crée un objet DomXsltStylesheet à partir du document XSL donné.

8.29.79 domxml_xslt_stylesheet_file() : Crée un objet DomXsltStylesheet à partir d'un document XSL dans un fichier

DomXsltStylesheet **domxml_xslt_stylesheet_file** (string xsl_file)

Crée un objet DomXsltStylesheet à partir du fichier XSL donné.

8.29.80 domxml_xslt_stylesheet() : Crée un objet DomXsltStylesheet à partir d'un document XSL dans une chaîne

DomXsltStylesheet **domxml_xslt_stylesheet** (string xsl_buf)

Crée un objet DomXsltStylesheet à partir d'une mémoire tampon XSL.

8.29.81 domxml_xslt_version() : Lit le numéro de version de la bibliothèque XSLT

int **domxml_xslt_version** (void)

Lit le numéro de version de la bibliothèque XSLT.

8.29.82 DomXsltStylesheet->process()() : Applique une transformation XSLT à un objet domdocument

DomXsltStylesheet DomDocument **process** (DomDocument xml_doc , array xslt_params , bool is_xpath_param , string profile_filename)

Applique une Transformation XSLT sur l'objet DomDocument donné.

8.29.83 DomXsltStylesheet->result_dump_file()() : Écrit le résultat d'une transformation XSLT dans un fichier

DomXsltStylesheet string **result_dump_file** (DomDocument xmldoc , string filename)

Comme DomXsltStylesheet->process() retourne toujours un document XML correctement formaté, quelque soit la méthode d'affichage qui a été déclarée dans <xsl:output> ou dans un attribut similaire, elle n'est pas très pratique si vous voulez produire des documents HTML 4 ou du texte.

Cette fonction, au contraire, utilise <xsl:output method="html|text"> et les autres directives de contrôle des sorties. Voyez l'exemple pour plus de détails.

8.29.84 DomXsltStylesheet->result_dump_mem()() : Écrit le résultat d'une transformation XSLT dans une chaîne

DomXsltStylesheet string **result_dump_mem** (DomDocument xmldoc)

Comme DomXsltStylesheet->process() retourne toujours un document XML correctement formaté, quelque soit la méthode d'affichage qui a été déclarée dans <xsl:output> ou dans un attribut similaire, elle n'est pas très pratique si vous voulez produire des documents HTML 4 ou du texte.

Cette fonction, au contraire, utilise <xsl:output method="html|text"> et les autres directives de contrôle des sorties. Voyez l'exemple pour plus de détails.

8.29.85 xpath_eval_expression() : Calcule un chemin XPath à partir d'une chaîne

XPathContext XPathObject **xpath_eval_expression** (string expression , domnode contextnode)
XPathObject **xpath_eval_expression** (XPathContext xpath_context , string expression , domnode contextnode)

Exemple avec xpath_eval_expression

```
<?php
include("example.inc");

if (!$dom = domxml_open_mem($xmlstr)) {
    echo "Erreur lors de l'analyse du document\n";
    exit;
}

$xmlpath = xpath_new_context($dom);
var_dump(xpath_eval_expression($xmlpath, '/chapter/@language'));
```

```
?>
```

L'exemple ci-dessus va afficher :

```
object(XPathObject)(2) {
["type"]=>
int(1)
["nodeset"]=>
array(1) {
[0]=>
object(domattribute)(5) {
["type"]=>
int(2)
["name"]=>
string(8) "language"
["value"]=>
string(2) "en"
[0]=>
int(7)
[1]=>
int(138004256)
}
}
}
```

Voir aussi [xpath_eval](#) .

8.29.86 `xpath_eval()` : Calcule un chemin XPath à partir d'une chaîne

```
XPathContext XPathObject xpath_eval ( string xpath_expression , domnode contextnode )
XPathObject xpath_eval ( XPathContext xpath_context , string xpath_expression , domnode contextnode )
```

Le paramètre optionnel `contextnode` sert à spécifier des requêtes XPath relatives.

Voir aussi [xpath_new_context](#) .

8.29.87 `xpath_new_context()` : Crée un nouveau contexte xpath

```
XPathContext xpath_new_context ( domdocument dom_document )
```

Voir aussi [xpath_eval](#) .

8.29.88 `xpath_register_ns_auto()` : Sauvegarde l'espace de nom donné dans le contexte XPath passé

```
bool xpath_register_ns_auto ( XPathContext xpath_context , object context_node )
```

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.29.89 `xpath_register_ns()` : Sauvegarde l'espace de nom donné dans le contexte XPath passé

`bool xpath_register_ns (XPathContext xpath_context , string prefix , string uri)`

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.29.90 `xptr_eval()` : Calcul un chemin XPtr à partir d'une chaîne

`XPathContext int xptr_eval (string eval_str , domnode contextnode) int xptr_eval (XPathContext xpath_context , string eval_str , domnode contextnode)`

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.29.91 `xptr_new_context()` : Crée un nouveau contexte XPath

`XPathContext xptr_new_context (void)`

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.30 Fonctions .NET

8.30.1 Introduction

Attention

Ce module est **EXPERIMENTAL** . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, **SANS PREAVIS!** Soyez-en conscient, et utilisez ce module à vos risques et périls.

- [Introduction](#)
- [dotnet_load](#)

8.31 Gestion des erreurs

8.31.1 Introduction

Ces fonctions permettent de gérer les erreurs et de les enregistrer. Vous pouvez définir les règles de traitement des erreurs et choisir la manière de les enregistrer : vous pouvez adapter le rapport d'erreurs à vos besoins.

Avec les fonctions d'enregistrements, vous pouvez envoyer directement les rapports à d'autres machines (ou même les envoyer par email à un pager), à l'historique système, ou encore sélectionner les erreurs les plus importantes et ne pas enregistrer les autres.

La fonction de niveau d'erreur vous permet de personnaliser le niveau et le type d'erreur noté : depuis les inoffensives alertes jusqu'aux erreurs personnalisées retournées par les fonctions.

8.31.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.31.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.31.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

Nom	Par défaut	Modifiable	Historique
error_reporting	NULL	PHP_INI_ALL	
display_errors	"1"	PHP_INI_ALL	
display_startup_errors	"0"	PHP_INI_ALL	Disponible depuis PHP 4.0.3.
log_errors	"0"	PHP_INI_ALL	
log_errors_max_len	"1024"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
ignore_repeated_errors	"0"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
ignore_repeated_source	"0"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
report_memleaks	"1"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
track_errors	"0"	PHP_INI_ALL	
html_errors	"1"	PHP_INI_ALL	PHP_INI_SYSTEM en PHP <=4.2.3. Disponible depuis PHP 4.0.2.
docref_root	""	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
docref_ext	""	PHP_INI_ALL	Disponible depuis PHP 4.3.2.
error_prepend_string	NULL	PHP_INI_ALL	
error_append_string	NULL	PHP_INI_ALL	
error_log	NULL	PHP_INI_ALL	

warn_plus_overloading	NULL	PHP_INI??
-----------------------	------	-----------

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [Directives du php.ini](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

error_reporting entier

Fixe le niveau d'erreur. Ce paramètre est un entier, représentant un champ de bits. Ajoutez les valeurs suivantes pour choisir le niveau que vous désirez, telles que décrites dans la section [Constantes pré-définies](#) , et dans le fichier php.ini . Pour modifier cette configuration durant l'exécution du script, utilisez la fonction [error_reporting](#) . Voyez aussi la directive [display_errors](#) .

En PHP 4 et PHP 5, la configuration par défaut est E_ALL & ~ E_NOTICE . Elle montre toutes les erreurs, sauf les E_NOTICE . Il est recommandé de les afficher durant le développement.

Note
Activer le rapport d'erreur de niveau E_NOTICE durant le développement a des avantages. En terme de débogage, les message d>alertes vous signalent des bogues potentiels dans votre code. Par exemple, l'utilisation de valeurs non initialisées est signalée. Il est aussi plus pratique pour trouver des coquilles, et, ainsi, gagner du temps. Les messages NOTICE vous signaleront aussi les mauvaises pratiques de codage. Par exemple \$arr[item] doit toujours être écrit \$arr['item'] car PHP va considérer "item" comme une constante, au premier abord. Si cette constante n'est pas définie, alors il va l'utiliser comme une chaîne.
Note
En PHP 5, un nouveau niveau d'erreur nommé E_STRICT est disponible. Comme E_STRICT n'est pas inclus sans E_ALL , vous devez explicitement activer ce niveau d'erreur. Activer E_STRICT pendant le développement peut être bénéfique. Les messages STRICT vous aideront à utiliser la dernière et meilleure suggestion de méthode de codage, par exemple, vous alertera de l'utilisation de fonctions non recommandées.

En PHP 3, la configuration par défaut est (E_ERROR | E_WARNING | E_PARSE) , ce qui correspond à la même configuration qu'en PHP 4. Notez toutefois que les constantes ne sont pas supportées dans le php3.ini de PHP 3, et que le niveau d'erreur doit être spécifié numériquement, c'est à dire 7 .

display_errors boolean

Cette directive détermine si les erreurs doivent être affichées à l'écran ou non.

Note
C'est une directive nécessaire en développement mais qui ne doit jamais être utilisée sur un système en production. (e.g. systèmes connectés à Internet).
Note
Bien que display_errors peut être défini en cours d'exécution (avec la fonction ini_set), il n'aura aucun effet si le script a des erreurs fatales, car l'action désirée au moment de l'exécution ne sera pas exécutée.

display_startup_errors boolean

Même si display_errors est activé, des erreurs peuvent survenir lors de la séquence de démarrage de PHP, et ces erreurs sont cachées. Avec cette option, vous pouvez les afficher, ce qui est recommandé pour le débogage. En dehors de ce cas, il est fortement recommandé de laisser display_startup_errors à off.

log_errors boolean

Indique où les messages d'erreur générés doivent être écrits : dans l'historique du serveur ou dans [error_log](#) . Cette fonction est spécifique aux serveurs.

Note

Il est recommandé d'utiliser l'historique d'erreur, plutôt que d'afficher les erreurs sur les sites de production.

`log_errors_max_len` entier

Configure la taille maximale des erreurs qui seront enregistrées dans l'historique, en kilo octets. Dans les informations de `error_log`, l'origine est ajoutée. La valeur par défaut est de 1024. 0 signifie qu'il n'y a pas de limite de taille. Cette longueur est appliqué pour enregistrer dans l'historique les erreurs, afficher les erreurs et également à `$php_errormsg`.

Lorsqu'un entier est utilisé, sa valeur est mesuré en octets. Vous pouvez également utiliser la notation sténographique comme décrit dans cette [entrée de la FAQ](#).

`ignore_repeated_errors` boolean

Ne pas enregistrer des messages répétitifs. Les erreurs répétées doivent survenir au même moment, à la même ligne et depuis le même fichier de script, jusqu'à ce que `ignore_repeated_source` soit mis à TRUE.

`ignore_repeated_source` boolean

Ignore la source du message lors des messages répétés. Lorsque vous avez configuré cette option à On, vous n'enregistrerez pas les erreurs répétées provenant de fichiers et lignes de code différents.

`report_memleaks` boolean

Si ce paramètre est mis à Off, alors les fuites de mémoires ne seront pas affichées (sur la sortie standard, stdout ou dans les logs). Cette option n'a d'effet que si PHP a été compilé avec l'option de débogage, et si `error_reporting` inclut E_WARNING dans sa liste.

`track_errors` boolean

Si cette option est activée, le dernier message d'erreur sera placé dans la variable globale `$php_errormsg`.

`html_errors` boolean

Désactive les balises HTML dans les messages d'erreurs. Le nouveau format d'erreurs HTML fournit des messages cliquables, qui redirige l'utilisateur vers la documentation de l'erreur ou de la fonction. Ces références sont affectées par `docref_root` et `docref_ext`.

`docref_root` string

Le nouveau format d'erreur contient une référence à une page décrivant l'erreur, ou la fonction ayant causé l'erreur. Pour le manuel, vous pouvez télécharger ce dernier dans votre langue, et configurer cette option pour qu'elle pointe sur lui. Si votre copie du manuel est accessible à '/manual/', vous pouvez simplement utiliser `docref_root=/manual/`. De plus, vous devez configurer `docref_ext` pour qu'elle corresponde aux extensions de votre manuel. `docref_ext=.html`. Il est possible d'utiliser des références externes. Par exemple, vous pouvez utiliser

`docref_root=http://manual/en/` ou

`docref_root="http://londonize.it/?how=url&theme=classic&filter=Landon&url=http%3A%2F%2F"`

La plupart du temps, vous utilisez l'option `docref_root` avec un slash à la fin ('/'). Mais ce n'est pas obligatoire, comme le montre le second exemple ci-dessus.

Note

Cette directive est destiné à vous aider dans votre développement en rendant facile la consultation de la description d'une fonction. Ne jamais l'utiliser sur un système de production (e.g. système connecté à Internet).

`docref_ext` string

Voir aussi `docref_root`.

Note
La valeur de <code>docref_ext</code> doit commencer par un point '.'.

`error_prepend_string` string

La chaîne à placer avant les messages d'erreur.

`error_append_string` string

La chaîne à placer après les messages d'erreur.

`error_log` string

Nom du fichier où seront enregistrées les erreurs. Le fichier doit être accessible en écriture par l'utilisateur exécutant le serveur web. Si la valeur spéciale `syslog` est utilisée, les erreurs seront envoyées au système d'historique du serveur. Sous Unix, cela correspond à `syslog(3)` et sous Windows NT, à l'historique d'événement.

L'historique n'est pas supporté sous Windows 95. Voir aussi : [syslog](#) .

`warn_plus_overloading` boolean

Si cette option est activée, PHP va afficher une alerte lorsque l'opérateur d'addition (+) est utilisé avec des chaînes de caractères. Cela peut aider à trouver les erreurs où le plus est utilisé comme opérateur de concaténation au lieu de point (.). Cette option n'existe pas en PHP 4.

8.31.5 Constantes pré-définies

Les constantes listées ici sont toujours disponibles dans PHP.

Note			
Vous pouvez utiliser ces constantes dans le fichier <code>php.ini</code> mais pas hors de PHP, comme dans le fichier <code>httpd.conf</code> , où vous devez utiliser les valeurs de champs de bits.			
Valeur	Constante	Description	Note
1	<code>E_ERROR</code> (entier)	Les erreurs sont aussi affichées par défaut, et l'exécution du script est interrompue. Elles indiquent des erreurs qui ne peuvent pas être ignorées, comme des problèmes d'allocation de mémoire, par exemple.	
2	<code>E_WARNING</code> (entier)	Les alertes sont affichées par défaut, mais n'interrompent pas l'exécution du script. Elles indiquent un problème qui doit être intercepté par le script durant l'exécution du script. Par exemple, appeler <code>ereg</code> avec une regex invalide.	
4	<code>E_PARSE</code> (entier)	Les erreurs d'analyse ne doivent être générées que par l'analyseur. Elles ne sont citées ici que dans le but d'être exhaustif.	
8	<code>E_NOTICE</code> (entier)	Les notices ne sont pas affichées par défaut, et indiquent que le script a rencontré quelque chose qui peut être une erreur, mais peut aussi être un événement normal dans la vie du script. Par exemple, essayer d'accéder à une valeur qui n'a pas été déclarée, ou appeler <code>stat</code> sur un fichier qui n'existe pas.	
16	<code>E_CORE_ERROR</code> (entier)	Elles sont similaires aux erreurs <code>E_ERROR</code> , mais elles sont générées par le code de PHP. Les fonctions ne doivent pas générer ce genre d'erreur.	depuis PHP 4 seulement
32	<code>E_CORE_WARNING</code> (entier)	Elles sont similaires à <code>E_WARNING</code> , mais elles sont générées par le code de PHP. Les fonctions	depuis PHP 4 seulement

		ne doivent pas générer ce genre d'erreur.	
64	E_COMPILE_ERROR (entier)	Elles sont similaires à E_ERROR , mais elles sont générées par Zend Scripting Engine. Les fonctions ne doivent pas générer ce genre d'erreur.	depuis PHP 4 seulement
128	E_COMPILE_WARNING (entier)	Elles sont similaires à E_WARNING , mais elles sont générées par Zend Scripting Engine. Les fonctions ne doivent pas générer ce genre d'erreur.	depuis PHP 4 seulement
256	E_USER_ERROR (entier)	E_USER_ERROR est comparable à E_ERROR . Elle est générée en PHP par l'utilisation de la fonction <code>trigger_error</code> . Les fonctions ne doivent pas générer ce genre d'erreur.	depuis PHP 4 seulement
512	E_USER_WARNING (entier)	E_USER_WARNING est comparable à E_WARNING . Elle est générée en PHP par l'utilisation de la fonction <code>trigger_error</code> . Les fonctions ne doivent pas générer ce genre d'erreur.	depuis PHP 4 seulement
1024	E_USER_NOTICE (entier)	E_USER_WARNING est comparable à E_NOTICE . Elle est générée en PHP par l'utilisation de la fonction <code>trigger_error</code> . Les fonctions ne doivent pas générer ce genre d'erreur.	depuis PHP 4 seulement
2047	E_ALL (entier)	Toutes les erreurs et alertes supportées sauf le niveau E_STRICT .	
2048	E_STRICT (entier)	Notices au moment de l'exécution. Permet d'obtenir des suggestions de PHP pour modifier votre code, assurant ainsi une meilleure interopérabilité et compatibilité de celui-ci.	PHP 5 seulement.

Les valeurs ci-dessus (numérique ou symbolique) sont utilisées pour constituer des champs de bits, qui spécifient le niveau de rapport d'erreur. Vous pouvez utiliser les opérateurs de bits pour combiner ces valeurs pour en faire des masques qui filtrent certaines erreurs. Notez bien que seuls '|', '~', '!', '^' et '&' seront compris dans le fichier php.ini , et que aucun opérateur logique ne sera compris en php3.ini .

8.31.6 Exemples

Ci-dessous, vous trouverez un exemple de gestion des erreurs par PHP. Il y est défini un gestionnaire d'erreur, qui enregistre les informations dans un fichier (au format XML), et envoie un courriel au développeur si l'erreur est critique.

Gestion d'erreurs avancées en PHP
<pre> <?php // Nous allons faire notre propre gestion error_reporting(0); // Fonction spéciale de gestion des erreurs function userErrorHandler(\$errno, \$errmsg, \$filename, \$linenum, \$vars) { // Date et heure de l'erreur \$dt = date("Y-m-d H:i:s (T)"); // Définit un tableau associatif avec les chaînes d'erreur // En fait, les seuls niveaux qui nous intéressent // sont E_WARNING, E_NOTICE, E_USER_ERROR, // E_USER_WARNING et E_USER_NOTICE \$errortype = array (E_ERROR => "Erreur", </pre>

```

        E_WARNING           => "Alerte",
        E_PARSE            => "Erreur d'analyse",
        E_NOTICE           => "Note",
        E_CORE_ERROR       => "Core Error",
        E_CORE_WARNING     => "Core Warning",
        E_COMPILE_ERROR    => "Compile Error",
        E_COMPILE_WARNING  => "Compile Warning",
        E_USER_ERROR       => "Erreur spécifique",
        E_USER_WARNING     => "Alerte spécifique",
        E_USER_NOTICE      => "Note spécifique",
        E_STRICT           => "Runtime Notice"
    );

    // Les niveaux qui seront enregistrés
    $user_errors = array(E_USER_ERROR, E_USER_WARNING, E_USER_NOTICE);

    $err = "<errorentry>\n";
    $err .= "\t<datetime>" . $dt . "</datetime>\n";
    $err .= "\t<errornum>" . $errno . "</errornum>\n";
    $err .= "\t<errortype>" . $errortype[$errno] . "</errortype>\n";
    $err .= "\t<errormsg>" . $errmsg . "</errormsg>\n";
    $err .= "\t<scriptname>" . $filename . "</scriptname>\n";
    $err .= "\t<scriptlinenum>" . $linenum . "</scriptlinenum>\n";

    if (in_array($errno, $user_errors)) {
        $err .= "\t<vartrace>".wddx_serialize_value($vars, "Variables")."</vartrace>\n";
    }
    $err .= "</errorentry>\n\n";

    // sauvegarde de l'erreur, et mail si c'est critique
    error_log($err, 3, "/usr/local/php4/error.log");
    if ($errno == E_USER_ERROR) {
        mail("phpdev@example.com", "Critical User Error", $err);
    }
}

function distance($vect1, $vect2)
{
    if (!is_array($vect1) || !is_array($vect2)) {
        trigger_error("Incorrect parameters, arrays expected", E_USER_ERROR);
        return NULL;
    }

    if (count($vect1) != count($vect2)) {
        trigger_error("Vectors need to be of the same size", E_USER_ERROR);
        return NULL;
    }

    for ($i=0; $i<count($vect1); $i++) {
        $c1 = $vect1[$i]; $c2 = $vect2[$i];
        $d = 0.0;
        if (!is_numeric($c1)) {
            trigger_error("Coordinate $i in vector 1 is not a number, using zero",
                E_USER_WARNING);
            $c1 = 0.0;
        }
        if (!is_numeric($c2)) {
            trigger_error("Coordinate $i in vector 2 is not a number, using zero",
                E_USER_WARNING);
            $c2 = 0.0;
        }
        $d += $c2*$c2 - $c1*$c1;
    }
    return sqrt($d);
}

Sold_error_handler = set_error_handler("userErrorHandler");

```

```
// constante non définie, qui génère une alerte
$t = I_AM_NOT_DEFINED;

// définition de quelques vecteurs
$a = array (2, 3, "foo");
$b = array (5.5, 4.3, -1.6);
$c = array (1, -3);

// génère une erreur utilisateur
$t1 = distance ($c, $b)."\n";

// génère une erreur utilisateur
$t2 = distance ($b, "i am not an array")."\n";

// Génère une alerte
$t3 = distance ($a, $b)."\n";

?>
```

8.31.7 Voir aussi

Voir aussi [syslog](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [Voir aussi](#)
- [debug_backtrace](#)
- [debug_print_backtrace](#)
- [error_log](#)
- [error_reporting](#)
- [restore_error_handler](#)
- [restore_exception_handler](#)
- [set_error_handler](#)
- [set_exception_handler](#)
- [trigger_error](#)
- [user_error](#)

8.31.9 debug_print_backtrace() : Affiche une backtrace

void **debug_print_backtrace** (void)

[debug_print_backtrace](#) affiche la backtrace de PHP. Elle affiche les appels aux fonctions, aux fichiers incluses / requis ainsi que le travail sur les [eval](#) .

8.31.10 error_log() : Stocke un message d'erreur

bool **error_log** (string message , int message_type , string destination , string extra_headers)

Envoie un message d'erreur à l'historique du serveur web, à un port TCP ou un fichier.

8.31.11 `error_reporting()` : Fixe le niveau de rapport d'erreurs PHP

int `error_reporting` (*int level*)

`error_reporting` modifie la directive `error_reporting` pendant l'exécution du script. PHP possède plusieurs niveaux d'erreurs, utiliser cette fonction configure ce niveau pendant la durée (d'exécution) de votre script.

8.31.12 `restore_error_handler()` : Réactive l'ancienne fonction de gestion des erreurs

bool `restore_error_handler` (void)

Utilisée après avoir modifié la fonction de gestion des erreurs, grâce à `set_error_handler`, `restore_error_handler` permet de réutiliser l'ancienne version de gestion des erreurs (qui peut être la fonction PHP par défaut, ou une autre fonction utilisateur).

8.31.13 `restore_exception_handler()` : Réactive l'ancienne fonction de gestion d'exceptions

bool `restore_exception_handler` (void)

`restore_exception_handler` est utilisé, après le changement de la fonction de gestion d'exceptions avec la fonction `set_exception_handler`, pour revenir à l'ancien gestionnaire d'exceptions (qui peut être la fonction interne ou une fonction définie par l'utilisateur).

8.31.14 `set_error_handler()` : Spécifie une fonction utilisateur comme gestionnaire d'erreurs

mixed `set_error_handler` (callback `error_handler`, *int error_types*)

`set_error_handler` choisit la fonction utilisateur `error_handler` pour gérer les erreurs dans un script.

`set_error_handler` peut être utilisé pour définir votre propre manière de gérer les erreurs durant l'exécution, par exemple pour une application dans laquelle vous devez nettoyer les données/fichiers lorsqu'une erreur survient ou lorsque vous devez déclencher une erreur sous certaines conditions (en utilisant `trigger_error`).

Il faut se rappeler que la fonction standard de traitement des erreurs de PHP est alors complètement ignorée. `error_reporting` n'aura plus d'effet, et votre fonction de gestion des erreurs sera toujours appelée. Vous pourrez toujours lire la valeur de l'erreur courante de `error_reporting` et faire réagir la fonction de gestion des erreurs en fonction. Cette remarque est notamment valable si la commande a été préfixée par `@`.

Notez aussi qu'il est alors confié à cette fonction de terminer le script (`die`) si nécessaire. Si la fonction de gestion des erreurs se termine normalement, l'exécution du script se poursuivra avec l'exécution de la prochaine commande.

Les types d'erreur suivants ne peuvent pas être gérés avec cette fonction : `E_ERROR` , `E_PARSE` , `E_CORE_ERROR` , `E_CORE_WARNING` , `E_COMPILE_ERROR` , `E_COMPILE_WARNING` ainsi que la plupart des `E_STRICT` d'un fichier lorsque `set_error_handler` est appelé.

Si une erreur survient avant que le script ne soit exécuté (par exemple un téléchargement de fichier), le gestionnaire d'erreurs personnalisé ne pourra pas être appelé, car il n'est pas encore enregistré.

8.31.15 `set_exception_handler()` : Définit une fonction utilisateur de gestion d'exceptions

string `set_exception_handler` (callback `exception_handler`)

`set_exception_handler` définit le gestionnaire d'exceptions par défaut si une exception n'est pas attrapée avec un block d'essai/d'attrappe. L'exécution sera stoppé après l'appel à la fonction `exception_handler` .

La fonction `exception_handler` doit être définie avant l'appel de la fonction `set_exception_handler` . Ce gestionnaire de fonction doit accepter un paramètre qui sera l'objet représentant l'exception qui vient d'être lancée.

8.31.16 `trigger_error()` : Déclenche une erreur utilisateur

bool `trigger_error` (string `error_msg` , int `error_type`)

`trigger_error` est utilisé pour déclencher une erreur utilisateur. Elle peut aussi être utilisée en conjonction avec un gestionnaire d'erreurs interne, ou un gestionnaire d'erreurs utilisateur qui a été choisi comme gestionnaire d'erreurs avec `set_error_handler` .

`trigger_error` est pratique lorsque vous devez générer une réponse particulière lors de l'exécution.

8.31.17 `user_error()` : Alias de `trigger_error`

Cette fonction est un alias de : `trigger_error` .

8.32 Exécution de programmes externes

8.32.1 Introduction

Ces fonctions fournissent la possibilité de passer directement des commandes au système, mais aussi de protéger le système des commandes passées.

8.32.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.32.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.32.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.32.5 Types de ressources

Cette extension ne définit aucune ressource.

8.32.6 Constantes pré-définies

Cette extension ne définit aucune constante.

8.32.7 Voir aussi

Ces fonctions sont complétées par l'opérateur guillemets obliques . De plus, lorsque le safe mode est activé, vous devez configurer l'option safe_mode_exec_dir .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Voir aussi](#)
- [escapeshellarg](#)
- [escapeshellcmd](#)
- [exec](#)
- [passthru](#)
- [proc_close](#)
- [proc_get_status](#)

- [proc_nice](#)
- [proc_open](#)
- [proc_terminate](#)
- [shell_exec](#)
- [system](#)

8.32.9 [escapeshellcmd\(\)](#) : Protège les caractères spéciaux du Shell

string **escapeshellcmd** (string *command*)

[escapeshellcmd](#) échappe tous les caractères de la chaîne *command* qui pourraient avoir une signification spéciale dans une commande Shell. Cette fonction permet de s'assurer que la commande sera correctement passée à l'exécuteur de commande Shell [exec](#) et [system](#) , ou encore à [guillemets obliques](#) .

Les caractères suivants seront échappés : #&;`|*?~<>^()[]{}\$\\ , \x0A et \xFF . ' et " ne sont échappés que s'ils sont par paire. Sous Windows, tous ces caractères ainsi que % sont remplacés par un espace.

8.32.10 [exec\(\)](#) : Exécute un programme externe

string **exec** (string *command* , array *output* , int *return_var*)

[exec](#) exécute la commande *command* .

8.32.11 [passthru\(\)](#) : Exécute un programme externe et affiche le résultat brut

void **passthru** (string *command* , int *return_var*)

[passthru](#) est similaire à la fonction [exec](#) car les deux exécutent la commande *command* . Si l'argument *return_var* est présent, le code de statut de réponse UNIX y sera placé. Cette fonction doit être utilisée de préférence aux commandes [exec](#) ou [system](#) lorsque le résultat attendu est de type binaire, et doit être passé tel quel à un navigateur. Une utilisation classique de cette fonction est l'exécution de l'utilitaire [pbmplus](#) qui peut retourner une image. En fixant le résultat du contenu (Content-Type) à image/gif puis en appelant [pbmplus](#) pour obtenir une image gif, vous pouvez créer des scripts PHP qui retournent des images.

8.32.12 [proc_close\(\)](#) : Ferme un processus ouvert par [proc_open](#) et retourne le code de sortie

int **proc_close** (resource *process*)

[proc_close](#) est similaire à [pclose](#) hormis le fait qu'elle fonctionne avec les processus ouverts par [proc_open](#) . [proc_close](#) attend que le processus *process* se termine, puis retourne son code de sortie. Si vous avez des pipes ouverts avec ce processus, il faut les fermer avec [fclose](#) avant d'appeler cette fonction pour éviter des verrouillages : le processus peut ne pas pouvoir sortir tant que les pipes sont ouverts.

8.32.13 `proc_get_status()` : Lit les informations concernant un processus ouvert par `proc_open`

array `proc_get_status` (resource process)

`proc_get_status` lit les données concernant le processus `process` créé avec la fonction `proc_open` .

8.32.14 `proc_nice()` : Change la priorité d'exécution du processus courant

bool `proc_nice` (int increment)

`proc_nice` modifie la priorité du processus courant par le paramètre spécifié `increment` . Un paramètre `increment` positif atténuera la priorité du processus courant, tandis qu'une valeur négative `increment` augmentera la priorité.

`proc_nice` n'est pas lié à `proc_open` et ses fonctions associées d'aucune façon.

8.32.15 `proc_open()` : Exécute une commande et ouvre les pointeurs de fichiers pour les entrées / sorties

resource `proc_open` (string cmd , array descriptorspec , array pipes , string cwd , array env , array other_options)

`proc_open` est similaire à `popen` mais fournit un plus grand degré de contrôle sur l'exécution du programme.

PHP 5 introduit le support `pty` pour les systèmes avec Unix98 `ptys`. Cela permet à votre script d'interagir avec des applications qui s'attendent à dialoguer avec un terminal. Un `pty` fonctionne comme un pipe, mais est bi-directionnel : il n'y a pas besoin de spécifier de mode lecture ou écriture. L'exemple ci-dessous montre comment utiliser un `pty`. Notez que tous les pointeurs de fichier ne doivent pas nécessairement dialoguer avec le `pty`. Notez aussi que seul un `pty` est créé, même si `pty` est spécifié 3 fois. Dans les futures versions de PHP il sera possible de faire plus que simplement lire et écrire dans un `pty`.

8.32.16 `proc_terminate()` : Termine un processus ouvert par `proc_open`

int `proc_terminate` (resource process , int signal)

`proc_terminate` envoie un signal au processus `process` (créé avec `proc_open`) pour lui indiquer qu'il doit se terminer. `proc_terminate` se termine immédiatement après, et n'attend pas l'arrêt réel du processus.

Le paramètre optionnel `signal` n'est utile que sur les plates-formes POSIX : vous pouvez alors spécifier un signal à envoyer au processus, en utilisant l'appel système `kill(2)` . La valeur par défaut est alors `SIGTERM` .

`proc_terminate` vous permet de conclure un processus, et de continuer les autres tâches. Vous pouvez tester la présence de votre processus en utilisant la fonction `proc_get_status` .

8.32.17 `shell_exec()` : Exécute une commande via le Shell et retourne le résultat sous forme de chaîne

string **shell_exec** (string *cmd*)

`shell_exec` est identique aux guillemets obliques .

8.32.18 `system()` : Exécute un programme externe et affiche le résultat

string **system** (string *command* , int *return_var*)

`system` est la version PHP de la fonction C qui exécute la commande *command* et retourne le résultat.

L'appel à la fonction `system` tente également d'actualiser le buffer de sortie du serveur web après chaque ligne de sortie si PHP fonctionne en tant que module serveur.

Si vous devez exécuter une commande et récupérer tout le résultat sans aucune intervention, utilisez la fonction `passthru` .

8.33 EXIF

8.33.1 Introduction

Avec l'extension EXIF, vous pouvez travailler avec les meta-données des images. Par exemple, vous pouvez utiliser les fonctions EXIF pour lire les meta-données d'une image prise avec un appareil photographique numérique, et placées dans les en-têtes des images JPEG et TIFF .

8.33.2 Pré-requis

PHP doit être compilé avec l'option `--enable-exif` . PHP n'a pas besoin de bibliothèque supplémentaire pour faire fonctionner cette extension. Les utilisateurs Windows doivent aussi avoir activé l'extension [mbstring](#) .

8.33.3 Installation

Pour activer le support EXIF en PHP, il suffit d'ajouter l'option de compilation `--enable-exif` .

Les utilisateurs Windows doivent s'assurer que les bibliothèques DLL `php_mbstring.dll` et `php_exif.dll` sont spécifiées dans le fichier `php.ini` . La bibliothèque `php_mbstring.dll` doit être chargée **avant** la bibliothèque `php_exif.dll` : pensez à ajuster votre `php.ini` .

8.33.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

EXIF supporte automatiquement la conversion depuis Unicode et JIS, lorsque le module [mbstring](#) est compilé avec PHP. Cela se fait en décodant le commentaire avec le jeu de caractères spécifié. Le résultat est ensuite codé avec un autre jeu de caractères, compatible avec la sortie HTTP .

Nom	Par défaut	Modifiable	Historique
<code>exif.encode_unicode</code>	"ISO-8859-15"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
<code>exif.decode_unicode_motorola</code>	"UCS-2BE"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
<code>exif.decode_unicode_intel</code>	"UCS-2LE"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
<code>exif.encode_jis</code>	""	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
<code>exif.decode_jis_motorola</code>	"JIS"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
<code>exif.decode_jis_intel</code>	"JIS"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

`exif.encode_unicode` string

`exif.encode_unicode` définit la méthode de gestion des commentaires écrits en Unicode. La valeur par défaut est ISO-8859-15 , qui devrait fonctionner dans tous les pays non-asiatiques. Cette directive peut être laissée vide, ou prendre un des jeux de caractères supporté par `mbstring`. Si elle est vide, le jeu de caractères interne de `mbstring` sera utilisé.

`exif.decode_unicode_motorola` string
`exif.decode_unicode_motorola` définit le jeu de caractères de remplacement pour les commentaires utilisateurs écrits en Unicode, si l'ordre des bits est celui de Motorola (big-endian). Cette directive ne peut être laissée vide, et doit être un des jeux de caractères supportés par l'extension `mbstring`. La valeur par défaut est UCS-2BE .

`exif.decode_unicode_intel` string
`exif.decode_unicode_intel` définit le jeu de caractères de remplacement pour les commentaires utilisateurs écrits en Unicode, si l'ordre des bits est celui de Intel (little-endian). Cette directive ne peut être laissée vide, et doit être un des jeux de caractères supportés par l'extension `mbstring`. La valeur par défaut est UCS-2LE .

`exif.encode_jis` string
`exif.encode_jis` définit la méthode de gestion des commentaires écrits en caractères JIS. La valeur par défaut est une chaîne vide, qui fait que le jeu de caractères interne de `mbstring` est utilisé.

`exif.decode_jis_motorola` string
`exif.decode_jis_motorola` définit le jeu de caractères de remplacement pour les commentaires utilisateurs écrits en JIS, si l'ordre des bits est celui de Motorola (big-endian). Cette directive ne peut être laissée vide, et doit être un des jeux de caractères supportés par l'extension `mbstring`. La valeur par défaut est JIS .

`exif.decode_jis_intel` string
`exif.decode_jis_intel` définit le jeu de caractères de remplacement pour les commentaires utilisateurs écrits en JIS, si l'ordre des bits est celui de Intel (little-endian). Cette directive ne peut être laissée vide, et doit être un des jeux de caractères supportés par l'extension `mbstring`. La valeur par défaut est JIS .

8.33.5 Types de ressources

Cette extension ne définit aucune ressource.

8.33.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

`EXIF_USE_MBSTRING` (entier)

La fonction `exif_imagetype` liste plusieurs constantes connexes.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [exif_imagetype](#)
- [exif_read_data](#)
- [exif_tagname](#)
- [exif_thumbnail](#)
- [read_exif_data](#)

8.33.8 `exif_read_data()` : Lit les en-têtes EXIF dans les images JPEG ou TIFF

array `exif_read_data` (string filename , string sections , bool arrays , bool thumbnail)

`exif_read_data` lit les en-têtes EXIF des images JPEG et TIFF . Avec cette fonction, vous pouvez lire les données méta générées par les appareils photos numériques.

Les en-têtes EXIF tendent à être présents dans les images JPEG/TIFF générées par les appareils photos numériques, mais malheureusement, chaque appareil photo numérique a une idée différente de la façon dont leurs images doivent être marquées, donc, vous ne pouvez pas toujours compter sur un en-tête EXIF spécifique, bien que présent.

Les paramètres Height et Width sont calculés de la même façon que pour la fonction `getimagesize` , donc leurs valeurs ne feront parties d'aucun en-tête retourné. De même, l'index html est la représentation textuelle de la hauteur/largeur utilisée dans une balise image HTML classique.

Lorsqu'un en-tête EXIF contient une note de Copyright, cet en-tête peut alors contenir lui-même deux valeurs. Comme cette solution est incohérente avec les standards EXIF 2.10, la section COMPUTED retournera les deux en-têtes, Copyright.Photographer et Copyright.Editor , tandis que les sections IFD0 contiennent le tableau d'octets avec des caractères NULL pour séparer les deux entrées ; ou bien, juste la première entrée si le type de données était erroné (comportement par défaut de EXIF). La section COMPUTED va aussi contenir une entrée Copyright , qui sera soit la chaîne originale de copyright, soit une liste de valeurs séparées par des virgules de photos et de copyright de l'auteur.

La balise UserComment présente le même problème que la balise Copyright. Elle peut stocker deux valeurs : en premier, le jeu de caractères utilisé, puis la valeur elle-même. Si c'est le cas, la section IFD contiendra uniquement le jeu de caractères, ou bien un tableau d'octets. La section COMPUTED va stocker les deux entrées UserCommentEncoding et UserComment . L'index UserComment est disponible dans les deux cas, et il est préférable de l'utiliser, plutôt que la valeur de la section IFD0 .

`exif_read_data` valide les données des balises EXIF en accord avec la spécification EXIF (<http://exif.org/Exif2-2.PDF> , page 20).

Note
Windows ME/XP peuvent endommager les en-têtes EXIF lors de la connexion à la caméra. Plus d'informations disponibles sur http://www.canon.co.jp/Imaging/NOTICE/011214-e.html .

8.33.9 `exif_tagname()` : Lit le nom d'en-tête EXIF d'un index

string `exif_tagname` (string index)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.33.10 `exif_thumbnail()` : Récupère la miniature d'une image TIFF ou JPEG

string `exif_thumbnail` (string filename , int width , int height , int imagetype)

exif_thumbnail lit la miniature de l'image TIFF ou JPEG dans le fichier filename .

Si vous voulez afficher des miniatures avec cette fonction, vous devez envoyer le bon type MIME avec la fonction header .

Il est possible que la fonction exif_thumbnail n'arrive pas à créer l'image mais peut déterminer sa taille. Dans ce cas, la fonction retourne FALSE mais les paramètres width et height sont définis.

8.33.11 read_exif_data() : Alias de exif_read_data

Cette fonction est un alias de : exif_read_data .

8.34 Fonctions de monitoring d'altération de fichiers

8.34.1 Introduction

FAM surveille les fichiers et les dossiers, alertant les applications impliquées de changement. Pour plus d'informations sur FAM, voyez le site <http://oss.sgi.com/projects/fam/>.

Un script PHP peut spécifier une liste de fichiers à FAM pour les surveiller en utilisant les fonctions fournies par cette extension.

Le processus FAM est démarré lorsque la première connexion d'une application est ouverte. Il se termine après la fermeture de toutes les connexions.

Note
Cette extension a été déplacée dans le module <u>PECL</u> et ne sera plus intégrée dans PHP à partir de PHP 5.1.0.
Note
Cette extension n'est pas disponible sur les plates-formes Windows.

8.34.2 Pré-requis

Cette extension utilise les fonctions de la bibliothèque FAM, développée par SGI. Vous devez la télécharger et installer la bibliothèque FAM.

8.34.3 Installation

8.34.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.34.5 Types de ressources

8.34.5.1 Ressources FAM

8.34.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Constante	Description
FAMChanged (<u>entier</u>)	Une valeur qui a été modifiée pour un fichier ou un dossier. Elle est accessible via <code>fstat(1)</code> .
FAMDeleted (<u>entier</u>)	Un fichier ou un dossier a été effacé.
FAMStartExecuting (<u>entier</u>)	Un fichier exécutable a été lancé.

FAMStopExecuting (<u>entier</u>)	Un fichier exécutable a fini de s'exécuter.
FAMCreated (<u>entier</u>)	Un fichier a été créé dans un dossier.
FAMMoved (<u>entier</u>)	Cet événement ne se produit jamais.
FAMAcknowledge (<u>entier</u>)	Evénement survenu en réponse à <u>fam_cancel_monitor</u> .
FAMExists (<u>entier</u>)	Un événement survenu suite à une commande de surveillance d'un fichier ou d'un dossier. Lorsque le dossier est sous surveillance, un événement est généré pour ce dossier et tous ses fichiers.
FAMEndExist (<u>entier</u>)	Un événement après le dernier événement.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [fam_cancel_monitor](#)
- [fam_close](#)
- [fam_monitor_collection](#)
- [fam_monitor_directory](#)
- [fam_monitor_file](#)
- [fam_next_event](#)
- [fam_open](#)
- [fam_pending](#)
- [fam_resume_monitor](#)
- [fam_suspend_monitor](#)

8.34.8 [fam_close\(\)](#) : Ferme la connexion FAM

void **fam_close** (resource fam)

[fam_close](#) ferme une connexion au service FAM, précédemment ouverte en utilisant [fam_open](#) .

8.34.9 [fam_monitor_collection\(\)](#) : Monitore les changements d'une collection de fichiers dans un dossier

resource **fam_monitor_collection** (resource fam , string dirname , int depth , string mask)

[fam_monitor_collection](#) monitore une collection de fichiers dans le dossier dirname . Les fichiers à surveiller sont indiqués par le chemin vers le dossier dirname , la profondeur maximale depth à partir de ce dossier et une pattern shell mask pour filtrer les fichiers à surveiller.

Une événement FAM sera généré lorsque le status d'un fichier changera. Les codes des événements possibles sont décrit en détail dans la partie [constantes](#) de cette section.

Voir aussi [fam_monitor_file](#) , [fam_monitor_directory](#) , [fam_cancel_monitor](#) , [fam_suspend_monitor](#) et [fam_resume_monitor](#) .

8.34.10 **fam_monitor_directory()** : Monitore un dossier pour les changements

resource **fam_monitor_directory** (resource fam , string dirname)

fam_monitor_directory monitore un dossier `dirname` et tous ses fichiers. Un événement FAM sera généré à chaque fois que le status du dossier (i.e. le résultat de la fonction stat sur ce dossier) ou de son contenu (i.e. le résultat de readdir) changera.

Les codes des événements possibles sont décrit en détail dans la partie constantes de cette section.

Voir aussi fam_monitor_file , fam_monitor_collection , fam_cancel_monitor , fam_suspend_monitor et fam_resume_monitor .

8.34.11 **fam_monitor_file()** : Monitore un fichier régulier pour les changements

resource **fam_monitor_file** (resource fam , string filename)

fam_monitor_file met en place le monitoring pour un fichier. Un événement FAM sera généré à chaque fois que le status du fichier (i.e. le résultat de la fonction stat sur ce fichier) changera.

Les codes des événements possibles sont décrit en détail dans la partie constantes de cette section.

Voir aussi fam_monitor_directory , fam_monitor_collection , fam_cancel_monitor , fam_suspend_monitor et fam_resume_monitor .

8.34.12 **fam_next_event()** : Retourne le prochain événement FAM en attente

array **fam_next_event** (resource fam)

fam_next_event retourne l'événement FAM suivant dans la file d'attente. Cette fonction bloquera tant qu'un événement sera disponible, ce qui peut être vérifié en appelant fam_pending .

fam_next_event retournera un tableau contenant un code événement FAM dans l'entrée 'code', le chemin vers le fichier ciblé dans l'entrée 'filename' et optionnellement un nom d'hôte dans l'entrée 'hostname'.

Les codes événements possibles sont détaillés dans la partie constantes de cette section.

Voir aussi fam_pending .

8.34.13 **fam_open()** : Ouvre une connexion vers le démon FAM

resource **fam_open** (string appname)

fam_open ouvre une connexion vers le démon du service FAM. Le paramètre optionnel appname est une chaîne de caractères identifiant l'application pour les journaux de loggs.

Voir aussi fam_close .

8.34.14 fam_pending() : Vérifie si des événements FAM sont disponibles

int **fam_pending** (resource fam)

fam_pending retourne une valeur différente de zéro si des évènements sont disponibles pour être récupérés avec fam_next_event .

Voir aussi fam_next_event .

8.34.15 fam_resume_monitor() : Reprend un monitoring suspendu

bool **fam_resume_monitor** (resource fam , resource fam_monitor)

fam_resume_monitor reprend le monitoring d'une ressource précédemment suspendu avec fam_suspend_monitor .

Voir aussi fam_suspend_monitor .

8.34.16 fam_suspend_monitor() : Suspend temporairement le monitoring

bool **fam_suspend_monitor** (resource fam , resource fam_monitor)

fam_suspend_monitor suspend temporairement le monitoring d'une ressource précédemment ouverte avec une fonction FAM. Le monitoring peut être repris par la suite à l'aide de fam_resume_monitor sans avoir à en recréer un nouveau.

Voir aussi fam_resume_monitor et fam_cancel_monitor .

8.35 FrontBase

8.35.1 Introduction

Ces fonctions vous permettent d'accéder aux serveurs SQL FrontBase. Plus de détails sur le serveur FrontBase sont disponibles sur le site de <http://www.frontbase.com/>.

La documentation complète de FrontBase est disponible à <http://www.frontbase.com/cgi-bin/WebObjects/FrontBase.woa/wa/productsPage?currentPage=Documentation>.

Frontbase est supporté depuis PHP 4.0.6.

8.35.2 Pré-requis

Vous devez installer un serveur de bases de données FrontBase ou, tout au moins, les bibliothèques clientes fbsql, pour accéder à ces fonctions. Vous pouvez accéder au site de FrontBase à l'adresse <http://www.frontbase.com/>.

8.35.3 Installation

Pour pouvoir utiliser ces fonctions, vous devez compiler PHP avec le support fbsql en utilisant l'option de configuration `--with-fbsql`. Si vous utilisez cette option sans spécifier le chemin jusqu'au dossier d'installation fbsql, PHP recherchera les bibliothèques du client fbsql dans les dossiers habituels, sur votre système. Les utilisateurs qui ont installé FrontBase dans un dossier non standard doivent spécifier le chemin comme ceci : `--with-fbsql=/path/to/fbsql`. Cela va indiquer à PHP le bon emplacement des bibliothèques de FrontBase, et éviter les conflits.

8.35.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable	Historique
<code>fbsql.allow_persistent</code>	"1"	PHP_INI_SYSTEM	Disponible depuis PHP 4.2.0.
<code>fbsql.generate_warnings</code>	"0"	PHP_INI_SYSTEM	Disponible depuis PHP 4.0.6.
<code>fbsql.autocommit</code>	"1"	PHP_INI_SYSTEM	Disponible depuis PHP 4.0.6.
<code>fbsql.max_persistent</code>	"-1"	PHP_INI_SYSTEM	Disponible depuis PHP 4.0.6.
<code>fbsql.max_links</code>	"128"	PHP_INI_SYSTEM	Disponible depuis PHP 4.0.6.
<code>fbsql.max_connections</code>	"128"	PHP_INI_SYSTEM	Disponible depuis PHP 4.0.6.
<code>fbsql.max_results</code>	"128"	PHP_INI_SYSTEM	Disponible depuis PHP 4.0.6.
<code>fbsql.batchSize</code>	"1000"	PHP_INI_SYSTEM	
<code>fbsql.default_host</code>	NULL	PHP_INI_SYSTEM	Disponible depuis PHP 4.0.6.
<code>fbsql.default_user</code>	"_SYSTEM"	PHP_INI_SYSTEM	Disponible depuis PHP 4.0.6.
<code>fbsql.default_password</code>	""	PHP_INI_SYSTEM	Disponible depuis PHP 4.0.6.
<code>fbsql.default_database</code>	""	PHP_INI_SYSTEM	Disponible depuis PHP 4.0.6.
<code>fbsql.default_database_password</code>	""	PHP_INI_SYSTEM	Disponible depuis PHP 4.0.6.

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [Directives du php.ini](#) .

8.35.5 Types de ressources

8.35.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

FBSQL_ASSOC ([entier](#))
FBSQL_NUM ([entier](#))
FBSQL_BOTH ([entier](#))
FBSQL_LOCK_DEFERRED ([entier](#))
FBSQL_LOCK_OPTIMISTIC ([entier](#))
FBSQL_LOCK_PESSIMISTIC ([entier](#))
FBSQL_ISO_READ_UNCOMMITTED ([entier](#))
FBSQL_ISO_READ_COMMITTED ([entier](#))
FBSQL_ISO_REPEATABLE_READ ([entier](#))
FBSQL_ISO_SERIALIZABLE ([entier](#))
FBSQL_ISO_VERSIONED ([entier](#))
FBSQL_UNKNOWN ([entier](#))
FBSQL_STOPPED ([entier](#))
FBSQL_STARTING ([entier](#))
FBSQL_RUNNING ([entier](#))
FBSQL_STOPPING ([entier](#))
FBSQL_NOEXEC ([entier](#))
FBSQL_LOB_DIRECT ([entier](#))
FBSQL_LOB_HANDLE ([entier](#))

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [fbsql_affected_rows](#)
- [fbsql_autocommit](#)
- [fbsql_blob_size](#)
- [fbsql_change_user](#)
- [fbsql_clob_size](#)
- [fbsql_close](#)
- [fbsql_commit](#)
- [fbsql_connect](#)
- [fbsql_create_blob](#)
- [fbsql_create_clob](#)
- [fbsql_create_db](#)
- [fbsql_data_seek](#)
- [fbsql_database_password](#)
- [fbsql_database](#)
- [fbsql_db_query](#)
- [fbsql_db_status](#)
- [fbsql_drop_db](#)

- [fbsql_erro](#)
- [fbsql_error](#)
- [fbsql_fetch_array](#)
- [fbsql_fetch_assoc](#)
- [fbsql_fetch_field](#)
- [fbsql_fetch_lengths](#)
- [fbsql_fetch_object](#)
- [fbsql_fetch_row](#)
- [fbsql_field_flags](#)
- [fbsql_field_len](#)
- [fbsql_field_name](#)
- [fbsql_field_seek](#)
- [fbsql_field_table](#)
- [fbsql_field_type](#)
- [fbsql_free_result](#)
- [fbsql_get_autostart_info](#)
- [fbsql_hostname](#)
- [fbsql_insert_id](#)
- [fbsql_list_dbs](#)
- [fbsql_list_fields](#)
- [fbsql_list_tables](#)
- [fbsql_next_result](#)
- [fbsql_num_fields](#)
- [fbsql_num_rows](#)
- [fbsql_password](#)
- [fbsql_pconnect](#)
- [fbsql_query](#)
- [fbsql_read_blob](#)
- [fbsql_read_clob](#)
- [fbsql_result](#)
- [fbsql_rollback](#)
- [fbsql_select_db](#)
- [fbsql_set_lob_mode](#)
- [fbsql_set_password](#)
- [fbsql_set_transaction](#)
- [fbsql_start_db](#)
- [fbsql_stop_db](#)
- [fbsql_tablename](#)
- [fbsql_username](#)
- [fbsql_warnings](#)

8.35.8 [fbsql_autocommit\(\)](#) : Active ou désactive la validation automatique

bool **fbsql_autocommit** (resource link_identifiant , bool OnOff)

[fbsql_autocommit](#) retourne l'état courant de la validation automatique, pour la connexion link_identifiant . Si le paramètre onoff est fourni, et vaut TRUE , FBSQL va se mettre en mode d'auto-validation, et les requêtes seront validées automatiquement si aucune erreur n'est trouvée. Si le paramètre onoff est fourni, et vaut FALSE , FBSQL va se mettre en mode de validation manuelle, et les requêtes seront validées par l'appel de la fonction [fbsql_commit](#) ou annulées par [fbsql_rollback](#) .

Voir aussi [fbsql_commit](#) et [fbsql_rollback](#)

8.35.9 fbsql_blob_size() : Retourne la taille d'un BLOB

int **fbsql_blob_size** (string blob_handle , resource link_identifieur)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.35.10 fbsql_change_user() : Change le nom d'utilisateur de la session active

resource **fbsql_change_user** (string user , string password , string database , resource link_identifieur)

[fbsql_change_user](#) change le nom de l'utilisateur courant sur la session active courante, ou sur `link_identifieur` . Si une base de données est spécifiée avec le paramètre `database` , elle deviendra la base par défaut du nouvel utilisateur. Le nouvel utilisateur doit être spécifié par son login (`user`), et son mot de passe (`password`). Si l'identification échoue, la session courante restera ouverte.

8.35.11 fbsql_clob_size() : Retourne la taille d'un CLOB

int **fbsql_clob_size** (string clob_handle , resource link_identifieur)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.35.12 fbsql_close() : Ferme la connexion FrontBase

bool **fbsql_close** (resource link_identifieur)

[fbsql_close](#) retourne TRUE en cas de succès et FALSE en cas d'erreur.

[fbsql_close](#) ferme la connexion au serveur FrontBase associée à la ressource `link_identifieur` . Si `link_identifieur` est omis, c'est la dernière connexion ouverte qui sera fermée.

Utiliser [fbsql_close](#) n'est pas nécessaire, car les liens non persistants seront automatiquement fermés à la fin du script.

Exemple avec fbsql_close
--

<pre><?php \$link = fbsql_connect("localhost", "_SYSTEM", "secret") or die("Impossible de se connecter"); echo 'Connecté!'; fbsql_close(\$link); ?></pre>

Voir aussi [fbsql_connect](#) et [fbsql_pconnect](#) .

8.35.13 `fbsql_commit()` : Valide une transaction

bool `fbsql_commit` (*resource link_identif*ier)

`fbsql_commit` termine la transaction courante en sauvant toutes les insertions, modifications et effacements sur le serveur, puis en relachant tous les verrous qui ont été posés sur les tables. Cette commande n'est nécessaire que si FBSQL est en mode de validation manuelle.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `fbsql_autocommit` et `fbsql_rollback`

8.35.14 `fbsql_connect()` : Ouvre une connexion à un serveur FrontBase

resource `fbsql_connect` (*string hostname* , *string username* , *string password*)

`fbsql_connect` retourne une ressource de connexion positive en cas de succès, ou un message d'erreur en cas d'échec.

`fbsql_connect` établit une connexion avec un serveur FrontBase. Les valeurs suivantes sont utilisées, en cas d'omission : `hostname = ' NULL '`, `username = '_SYSTEM'` et `password = ''` (pas de mot de passe).

Si un deuxième appel est fait à `fbsql_connect` avec les mêmes arguments, une nouvelle connexion ne sera pas générée, mais la connexion déjà ouverte sera réutilisée, et retournée.

La connexion au serveur sera fermée dès la fin du script, à moins qu'elle ne soit explicitement terminée plus tôt, avec la fonction `fbsql_close` .

Exemple avec `fbsql_connect`

```
<?php
$link = fbsql_connect("localhost", "_SYSTEM", "secret")
      or die("Impossible de se connecter");
echo 'Connexion effectuée';
fbsql_close($link);
?>
```

Voir aussi `fbsql_pconnect` et `fbsql_close` .

8.35.15 `fbsql_create_blob()` : Crée un BLOB

string `fbsql_create_blob` (*string blob_data* , *resource link_identif*ier)

`fbsql_create_blob` retourne une ressource qui représente le nouveau BLOB.

`fbsql_create_blob` crée un blob à partir des données fournies par `blob_data` . La ressource retournée est nécessaire aux commandes de modification et d'insertion, pour sauver le BLOB dans la base de données.

Exemple avec `fbsql_create_blob`

```
<?php
$link = fbsql_pconnect ("localhost", "_SYSTEM", "secret")
      or die ("Impossible de se connecter");
```

```

$filename = 'blobfile.bin';
$fp = fopen($filename, "rb");
$blobdata = fread($fp, filesize($filename));
fclose($fp);
$blobHandle = fbsql_create_blob($blobdata, $link);
$sql = "INSERT INTO BLOB_TABLE (BLOB_COLUMN) VALUES ($blobHandle)";
$rs = fbsql_query($sql, $link);
?>

```

Voir aussi [fbsql_create_blob](#) , [fbsql_read_blob](#) , [fbsql_read_clob](#) et [fbsql_set_lob_mode](#) .

8.35.16 fbsql_create_clob() : Crée un CLOB

string **fbsql_create_clob** (string clob_data , resource link_identifieur)

[fbsql_create_clob](#) retourne une ressource qui représente le nouveau CLOB.

[fbsql_create_clob](#) crée un clob à partir des données fournies par clob_data . La ressource retournée est nécessaire aux commandes de modification et d'insertion, pour sauver le CLOB dans la base de données.

Exemple avec [fbsql_create_clob](#)

```

<?php
$link = fbsql_pconnect ("localhost", "_SYSTEM", "secret")
    or die ("Impossible de se connecter");
$filename = 'clob_file.txt';
$fp = fopen($filename, "rb");
$clobdata = fread($fp, filesize($filename));
fclose($fp);
$clobHandle = fbsql_create_clob($clobdata, $link);
$sql = "INSERT INTO CLOB_TABLE (CLOB_COLUMN) VALUES ($clobHandle)";
$rs = fbsql_query($sql, $link);
?>

```

Voir aussi [fbsql_create_blob](#) , [fbsql_read_blob](#) , [fbsql_read_clob](#) et [fbsql_set_lob_mode](#) .

8.35.17 fbsql_create_db() : Crée une base de données

bool **fbsql_create_db** (string database_name , resource link_identifieur , string database_options)

[fbsql_create_db](#) crée une nouvelle base de données, nommée database_name , sur le serveur repéré par la ressource link_identifieur .

Exemple avec [fbsql_create_db](#)

```

<?php
$link = fbsql_pconnect ("localhost", "_SYSTEM", "secret")
    or die ("Impossible de se connecter");
if (fbsql_create_db("my_db")) {
    echo "Base de données créée!\n";
} else {
    printf("Erreur de création de la base de données : %s\n", fbsql_error());
}
?>

```

Voir aussi [fbsql_drop_db](#) .

8.35.18 `fbsql_data_seek()` : Déplace le pointeur interne de résultat FBSQL

bool **fbsql_data_seek** (resource result_identifier , int row_number)

`fbsql_data_seek` déplace le pointeur interne de ligne dans le résultat de requête `result_identifier` jusqu'à la ligne `row_number` . Le prochain appel à `fbsql_fetch_row` retournera cette ligne.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Les lignes sont numérotées à partir de 0.

Exemple avec `fbsql_data_seek`

```
<?php
$link = fbsql_pconnect("localhost", "_SYSTEM", "secret")
    or die ("Impossible de se connecter");
fbsql_select_db("samp_db")
    or die ("Impossible de sélectionner une base");
$query = "SELECT last_name, first_name FROM friends;";
$result = fbsql_query($query)
    or die ("La requête est mauvaise");

// Lecture des lignes en ordre inverse
for ($i = fbsql_num_rows($result) - 1; $i >=0; $i--) {

    if (!fbsql_data_seek($result, $i)) {
        printf ("Impossible d'accéder à la ligne %d\n", $i);
        continue;
    }
    if(!($row = fbsql_fetch_object($result))) {
        continue;
    }

    echo $row->last_name . $row->first_name . "<br />\n";
}
fbsql_free_result($result);
?>
```

8.35.19 `fbsql_database_password()` : Modifie/lit le mot de passe dans une base FBSQL

string **fbsql_database_password** (resource link_identifier , string database_password)

`fbsql_database_password` retourne le mot de passe du serveur représenté par `link_identifier` .

`fbsql_database_password` modifie et lit le mot de passe de la base de données courante. Si le second paramètre `database_password` est fourni, `fbsql_database_password` remplacera le mot de passe courant par celui-ci. Si aucun serveur n'est précisé car le paramètre `link_identifier` n'est pas fourni, la dernière connexion ouverte sera utilisée. Si aucune connexion n'a été ouverte PHP essaiera d'en ouvrir une en appelant la fonction `fbsql_connect` , et en utilisant la connexion qui en résultera (si elle réussit).

Cette fonction ne modifie pas le mot de passe d'accès au serveur qui est stocké sur le serveur, et elle ne permet pas non plus d'aller lire ce mot de passe.

Exemple avec [fbsql_create_clob](#)

```
<?php
$link = fbsql_pconnect ("localhost", "_SYSTEM", "secret")
        or die ("Impossible de se connecter");
fbsql_database_password($link, "mot de passe secret");
fbsql_select_db($database, $link);
?>
```

Voir aussi [fbsql_select_db](#) , [fbsql_connect](#) et [fbsql_pconnect](#) .

8.35.20 [fbsql_database\(\)](#) : Lit ou définit le nom de la base de données utilisée avec une connexion

string **fbsql_database** (resource link_identifieur , string database)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.35.21 [fbsql_db_query\(\)](#) : Envoie une requête à la base FrontBase

resource **fbsql_db_query** (string database , string query , resource link_identifieur)

[fbsql_db_query](#) retourne une ressource positive représentant un résultat de requête en cas de succès, et FALSE en cas d'erreur.

[fbsql_db_query](#) sélectionne la base database et y exécute la requête query . Si le paramètre optionnel link_identifieur est spécifié, [fbsql_db_query](#) travaillera sur cette connexion. S'il est omis, [fbsql_db_query](#) essaiera d'utiliser la dernière connexion ouverte. Si aucune connexion n'a été ouverte, [fbsql_db_query](#) essaiera de se connecter automatiquement en appelant la fonction [fbsql_connect](#) , sans argument.

Voir aussi [fbsql_connect](#) .

8.35.22 [fbsql_db_status\(\)](#) : Lit le statut courant d'une base de données

int **fbsql_db_status** (string database_name , resource link_identifieur)

[fbsql_db_status](#) retourne un entier représentant le statut courant.

[fbsql_db_status](#) demande le statut de la base de données nommée database_name . Si le paramètre link_identifieur est omis, la connexion par défaut sera utilisée.

La valeur de l'entier peut être l'une des suivantes :

- FALSE - Le gestion de cet hôte était invalide. Cette erreur survient lorsque PHP se connecte directement à une base de données en utilisant un numéro de port. FBExec peut être disponible sur un serveur, mais aucune connexion n'a été faite.
- FBSQL_UNKNOWN - Le statut est inconnu.
- FBSQL_STOPPED - La base de données ne fonctionne pas. Utilisez [fbsql_start_db](#) pour démarrer la base.
- FBSQL_STARTING - La base de données démarre.

- FBSQL_RUNNING - La base de données fonctionne, et est disponible pour recevoir des requêtes SQL.
- FBSQL_STOPPING - La base de données s'arrête.
- FBSQL_NOEXEC - FBExec ne fonctionne pas sur le serveur, et il n'est pas possible d'obtenir le statut de la base de données.

Voir aussi [fbsql_start_db](#) et [fbsql_stop_db](#) .

8.35.23 fbsql_drop_db() : Supprime une base de données FrontBase

bool **fbsql_drop_db** (string database_name , resource link_identifier)

[fbsql_drop_db](#) essaie de supprimer la base de données database_name , sur la connexion représentée par link_identifier .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.35.24 fbsql_errno() : Retourne le code d'erreur FrontBase

int **fbsql_errno** (resource link_identifier)

[fbsql_errno](#) retourne le code d'erreur de la dernière connexion FrontBase, ou bien 0 (zéro) si aucune erreur n'est survenue.

Les erreurs générées par FrontBase ne sont pas automatiquement affichées comme alertes. Il faut utiliser la fonction [fbsql_errno](#) pour connaître leur code d'erreur. Notez que cette fonction ne retourne que le code d'erreur généré par la dernière fonction FrontBase (hormis [fbsql_error](#) et [fbsql_errno](#)) : si vous voulez repérer les erreurs, faites le dès que les fonctions ont été appelées.

Exemple avec [fbsql_errno](#)

```
<?php
fbsql_connect("marliesle");
echo fbsql_errno() . ' : ' . fbsql_error() . '<br />';
fbsql_select_db("nonexistentdb");
echo fbsql_errno() . ' : ' . fbsql_error() . '<br />';
$conn = fbsql_query("SELECT * FROM nonexistenttable;");
echo fbsql_errno() . ' : ' . fbsql_error() . '<br />';
?>
```

Voir aussi [fbsql_error](#) et [fbsql_warnings](#)

8.35.25 fbsql_error() : Retourne le message d'erreur FrontBase

string **fbsql_error** (resource link_identifier)

[fbsql_error](#) retourne le dernier message d'erreur généré par le serveur FrontBase, ou bien " (chaîne vide) si aucune erreur n'est survenue.

Les erreurs générées par FrontBase ne sont pas automatiquement affichées comme alertes. Il faut utiliser la fonction [fbsql_errno](#) pour connaître leur code d'erreur. Notez que cette fonction ne retourne que le code d'erreur généré par la dernière fonction FrontBase (hormis [fbsql_error](#) et [fbsql_errno](#)) : si vous voulez repérer les erreurs, faites le dès que les fonctions ont été appelées.

Exemple avec `fbsql_error`

```
<?php
fbsql_connect("marliesle");
echo fbsql_errno() . ' : ' . fbsql_error() . '<br />';
fbsql_select_db("nonexistentdb");
echo fbsql_errno() . ' : ' . fbsql_error() . '<br />';
$conn = fbsql_query("SELECT * FROM nonexistenttable;");
echo fbsql_errno() . ' : ' . fbsql_error() . '<br />';
?>
```

Voir aussi [fbsql_errno](#) et [fbsql_warnings](#)

8.35.26 `fbsql_fetch_array()` : Lit toute une ligne de résultat dans un tableau

array `fbsql_fetch_array` (resource result , int result_type)

`fbsql_fetch_array` retourne un tableau contenant la ligne courante du résultat result , ou FALSE s'il n'y a plus de ligne.

`fbsql_fetch_array` est une version améliorée de `fbsql_fetch_row` . En plus de stocker les données dans un tableau indexé, elle les stocke aussi sous forme de tableau associatif, dont les clés sont les noms des colonnes.

Si deux colonnes (ou plus) ont le même nom, la dernière colonne sera utilisée. Pour accéder aux autres colonnes de même nom, vous devez absolument utiliser les indices numériques.

Exemple avec `fbsql_fetch_array`

```
select t1.f1 as foo t2.f1 as bar from t1, t2
```

Il est important de noter que `fbsql_fetch_array` n'est pas significativement plus lente que `fbsql_fetch_row` , tandis qu'elle apporte un confort d'utilisation notable.

Le second argument optionnel result_type de `fbsql_fetch_array` est une constante qui peut prendre l'une des valeurs suivantes : FBSQL_ASSOC , FBSQL_NUM et FBSQL_BOTH .

Pour plus de détails, reportez-vous à [fbsql_fetch_row](#) et [fbsql_fetch_assoc](#) .

Exemple avec `fbsql_fetch_array`

```
<?php
fbsql_connect($host, $user, $password);
$result = fbsql_db_query("database","select user_id, fullname from table");
while ($row = fbsql_fetch_array($result)) {
    echo 'user_id : ' . $row["user_id"] . "<br />\n";
    echo 'user_id : ' . $row[0] . "<br />\n";
    echo 'Nom complet : ' . $row["fullname"] . "<br />\n";
    echo 'Nom complet : ' . $row[1] . "<br />\n";
}
fbsql_free_result($result);
?>
```

8.35.27 `fbsql_fetch_assoc()` : Lit toute une ligne de résultat dans un tableau associatif

array `fbsql_fetch_assoc` (resource result)

`fbsql_fetch_assoc` retourne un tableau associatif contenant la ligne courante du résultat `result`, ou `FALSE` s'il n'y a plus de lignes.

`fbsql_fetch_assoc` est équivalent à `fbsql_fetch_array` avec l'option `FBSQL_ASSOC`. Elle ne retourne qu'un tableau associatif. C'est le comportement initial de `fbsql_fetch_array`. Si vous avez aussi besoin des indices numériques, utilisez `fbsql_fetch_array`.

Si deux colonnes (ou plus) ont le même nom, la dernière colonne sera utilisée. Pour accéder aux autres colonnes de même nom, vous devez absolument utiliser la fonction `fbsql_fetch_array`.

Il est important de noter que `fbsql_fetch_assoc` n'est pas significativement plus lente que `fbsql_fetch_row`, tandis qu'elle apporte un confort d'utilisation notable.

Pour plus de détails, reportez-vous à `fbsql_fetch_row` et `fbsql_fetch_array`.

Exemple avec `fbsql_fetch_assoc`

```
<?php
fbsql_connect($host, $user, $password);
$result = fbsql_db_query("database","select * from table;");
while ($row = fbsql_fetch_assoc($result)) {
    echo $row["user_id"];
    echo $row["fullname"];
}
fbsql_free_result($result);
?>
```

8.35.28 `fbsql_fetch_field()` : Lit des informations sur une colonne dans un résultat, et retourne un objet

objet `fbsql_fetch_field` (*resource* `result`, *int* `field_offset`)

`fbsql_fetch_field` retourne un objet contenant les informations sur un champ, dans le résultat `result`.

`fbsql_fetch_field` sert à lire des informations sur les champs dans le résultat `result`. Si le second paramètre `field_offset` n'est pas spécifié, le champ suivant est lu.

Les propriétés de l'objet sont :

- `name` - Nom de colonne
- `table` - Nom de la table d'origine
- `max_length` - Taille maximale de la colonne
- `not_null` - 1 si la colonne ne peut être nulle
- `type` - Type de la colonne

Exemple avec `fbsql_fetch_field`

```
<?php
fbsql_connect($host, $user, $password)
    or die ("Impossible de se connecter");
$result = fbsql_db_query("database", "select * from table;");
    or die ("La requête a échoué");
// lire les données de colonnes
$i = 0;
while ($i < fbsql_num_fields($result)) {
    echo "Information de la colonne $i:<br />\n";
    $meta = fbsql_fetch_field($result);
```

```

if (!$meta) {
    echo "Aucune information disponible<br />\n";
}
echo "<pre>
max_length:    $meta->max_length
name:          $meta->name
not_null:      $meta->not_null
table:         $meta->table
type:          $meta->type
</pre>";
$i++;
}
fbsql_free_result($result);
?>

```

Voir aussi [fbsql_field_seek](#) .

8.35.29 [fbsql_fetch_lengths\(\)](#) : Lit la taille de chaque colonne d'un résultat

array [fbsql_fetch_lengths](#) (resource result)

[fbsql_fetch_lengths](#) retourne un tableau contenant les tailles maximales de chaque champ, dans la dernière ligne lue par [fbsql_fetch_row](#) ou FALSE en cas d'erreur.

[fbsql_fetch_lengths](#) stocke les tailles de chaque ligne de résultat retournée par [fbsql_fetch_row](#) , [fbsql_fetch_array](#) et [fbsql_fetch_object](#) dans un tableau à indices numériques, commençant à 0.

Voir aussi [fbsql_fetch_row](#) .

8.35.30 [fbsql_fetch_object\(\)](#) : Lit une ligne de résultat sous forme d'objet

object [fbsql_fetch_object](#) (resource result , int result_type)

[fbsql_fetch_object](#) retourne un objet dont les propriétés représentent les colonnes de la ligne à lire, dans le résultat result , ou FALSE s'il n'y a pas de ligne à lire.

[fbsql_fetch_object](#) est similaire à [fbsql_fetch_array](#) , à la différence qu'elle retourne un objet. Nous ne pouvons alors accéder aux données qu'avec les noms des colonnes, et sous la forme de membres d'objet, et non plus avec leurs index (les nombres ne peuvent représenter un membre d'objet).

En terme de vitesse, cette fonction est identique à [fbsql_fetch_array](#) et presque aussi rapide que [fbsql_fetch_row](#) (la différence n'est pas significative).

Exemple avec [fbsql_fetch_object](#)

```

<?php
fbsql_connect($host, $user, $password);
$result = fbsql_db_query("database", "select * from table;");
while ($row = fbsql_fetch_object($result)) {
    echo $row->user_id;
    echo $row->fullname;
}
fbsql_free_result($result);
?>

```

Voir aussi [fbsql_fetch_array](#) et [fbsql_fetch_row](#) .

8.35.31 [fbsql_fetch_row\(\)](#) : Lit une ligne de résultat sous forme de tableau numérique

array [fbsql_fetch_row](#) (resource result)

[fbsql_fetch_row](#) retourne un tableau représentant la ligne courante dans le résultat result , ou FALSE s'il n'y a plus de ligne à lire.

[fbsql_fetch_row](#) lit une ligne de données dans le résultat result et crée un tableau indexé. Chaque colonne est stockée dans une case du tableau, dans le même ordre que dans le résultat. Les indices commencent à 0.

Le prochain appel à [fbsql_fetch_row](#) va lire la prochaine ligne, ou bien retourner FALSE s'il n'y a plus de ligne à lire.

Voir aussi [fbsql_fetch_array](#) , [fbsql_fetch_object](#) , [fbsql_data_seek](#) , [fbsql_fetch_lengths](#) et [fbsql_result](#) .

8.35.32 [fbsql_field_flags\(\)](#) : Lit les options associées à une colonne de résultat

string [fbsql_field_flags](#) (resource result , int field_offset)

[fbsql_field_flags](#) retourne les options du champ field_offset , dans le résultat result . Les options sont retournées sous la forme d'un seul mot par option, séparées par des espaces, de façon à faciliter la manipulation avec [explode](#) .

8.35.33 [fbsql_field_len\(\)](#) : Retourne la taille d'un champ

int [fbsql_field_len](#) (resource result , int field_offset)

[fbsql_field_len](#) retourne la taille du champ field_offset dans le résultat result .

8.35.34 [fbsql_field_name\(\)](#) : Lit le nom d'un champ

string [fbsql_field_name](#) (resource result , int field_index)

[fbsql_field_name](#) retourne le nom du champ numéro field_index dans le résultat result . field_index est le résultat de [fbsql_query](#) et field_index est l'index numérique du champ.

Note
field_index commence à 0.
e.g. L'index du troisième champ est 2, et l'index du quatrième champ est 3...
Exemple avec fbsql_field_name
<?php

```
// La table utilisateur est constituée de trois colonnes
//  user_id
//  username
//  password
$res = fbsql_db_query("users", "select * from users;", $link);
echo fbsql_field_name($res, 0) . "\n";
echo fbsql_field_name($res, 2);
?>
```

L'exemple ci-dessus va afficher :

```
user_id
password
```

8.35.35 fbsql_field_seek() : Déplace le pointeur de résultat

bool **fbsql_field_seek** (resource result , int field_offset)

fbsql_field_seek place le pointeur de colonne à la colonne field_offset . Si ce paramètre est omis, fbsql_fetch_field retourne le numéro de la colonne courante.

Voir aussi fbsql_fetch_field .

8.35.36 fbsql_field_table() : Lit le nom de la table d'origine d'un champ

string **fbsql_field_table** (resource result , int field_offset)

fbsql_field_table retourne le nom de la table d'où est issu le champ d'indice field_offset . Les numéros de colonnes commencent à 0.

8.35.37 fbsql_field_type() : Lit le type d'une colonne

string **fbsql_field_type** (resource result , int field_offset)

fbsql_field_type est similaire à la fonction fbsql_field_name . Les arguments sont identiques, mais le type du champ est retourné. Il peut valoir "int", "real", "string", "blob" ou d'autres valeurs, comme décrit dans la documentation FrontBase .

Exemple avec fbsql_field_type

```
<?php
fbsql_connect("localhost:3306");
fbsql_connect("localhost", "_SYSTEM", "");
$result = fbsql_query("SELECT * FROM onek;");
$fields = fbsql_num_fields($result);
$rows   = fbsql_num_rows($result);
$i = 0;
$table = fbsql_field_table($result, $i);
echo "Votre table '" . $table . "' a " . $fields . " colonnes et " . $rows . " lignes <br />";
echo 'La table dispose des champs suivants <br />';
while ($i < $fields) {
    $type = fbsql_field_type ($result, $i);
    $name = fbsql_field_name ($result, $i);
    $len  = fbsql_field_len  ($result, $i);
    $flags = fbsql_field_flags($result, $i);
```

```

    echo $type . ' ' . $name . ' ' . $len . ' ' . $flags . '<br />';
    $i++;
}
fbsql_close();
?>

```

8.35.38 fbsql_free_result() : Libère le résultat de la mémoire

bool **fbsql_free_result** (resource result)

fbsql_free_result va libérer toute la mémoire utilisée par le résultat associé à la ressource result .

fbsql_free_result n'a besoin d'être appelé que si vous craignez que votre script ne consomme trop de mémoire, lorsqu'une requête retourne de très grands résultats. Toutes les ressources mémoire utilisées par le script sont de toute manière libérées à la fin du script.

8.35.39 fbsql_get_autostart_info() : Pas de documentation

array **fbsql_get_autostart_info** (resource link_identifieur)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.35.40 fbsql_hostname() : Lit ou définit le nom de l'hôte à utiliser avec une connexion

string **fbsql_hostname** (resource link_identifieur , string host_name)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.35.41 fbsql_insert_id() : Lit le dernier identifiant généré par une requête INSERT

int **fbsql_insert_id** (resource link_identifieur)

fbsql_insert_id retourne l'identifiant généré par la colonne de type DEFAULT UNIQUE, lors de la dernière requête INSERT, avec la connexion link_identifieur . Si link_identifieur est omis, la dernière connexion ouverte est utilisée.

fbsql_insert_id retournera 0 si la dernière requête n'a pas généré de valeur dans la colonne DEFAULT UNIQUE. Si vous devez sauver cette valeur pour plus tard, n'oubliez pas d'appeler fbsql_insert_id tout de suite après la requête qui a généré cette valeur.

Note

La valeur de la fonction FrontBase SQL " LAST_INSERT_ID() " retourne toujours la dernière valeur générée par DEFAULT UNIQUE et n'est jamais annulée entre les requêtes.

8.35.42 `fbsql_list_dbs()` : Liste les bases de données FBSQL

resource `fbsql_list_dbs` (resource *link_identifieur*)

`fbsql_list_dbs` retourne un résultat contenant la liste des bases de données disponibles sur le serveur *link_identifieur* . Utilisez la fonction `fbsql_tablename` pour passer en revue ce résultat.

Exemple avec `fbsql_list_dbs`

```
<?php
$link = fbsql_connect('localhost', 'myname', 'secret');
$db_list = fbsql_list_dbs($link);
while ($row = fbsql_fetch_object($db_list)) {
    echo $row->Database . "\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
database1
database2
database3
...
```

Note

L'exemple ci-dessus peut aussi bien fonctionner avec la fonction `fbsql_fetch_row` ou toute autre similaire.

8.35.43 `fbsql_list_fields()` : Liste les champs d'un résultat FrontBase

resource `fbsql_list_fields` (string *database_name* , string *table_name* , resource *link_identifieur*)

`fbsql_list_fields` lit les informations à propos de la table *table_name* , dans la base de données *table_name* , sur la connexion *link_identifieur* . Un résultat de requête est retourné, et pourra être utilisé avec les fonctions `fbsql_field_flags` , `fbsql_field_len` , `fbsql_field_name` et `fbsql_field_type` .

Un identifiant de résultat est une ressource PHP, représentée par un entier positif. `fbsql_list_fields` retourne -1 en cas d'erreur. Une chaîne décrivant l'erreur sera alors placée dans la variable `$phperrormsg` . Un message d'erreur sera aussi affiché, à moins que la fonction n'ait été appelée avec l'opérateur de suppression des erreurs `@`.

Exemple avec `fbsql_list_fields`

```
<?php
$link = fbsql_connect('localhost', 'monnom', 'secret');
$fields = fbsql_list_fields("database1", "table1", $link);
$columns = fbsql_num_fields($fields);
for ($i = 0; $i < $columns; $i++) {
    echo fbsql_field_name($fields, $i) . "\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
field1
field2
field3
...
```

8.35.44 `fbsql_list_tables()` : Liste les tables dans une base de données FrontBase

resource `fbsql_list_tables` (string database , resource link_identifier)

`fbsql_list_tables` liste les tables dans la base de données database , et retourne un résultat, tout comme `fbsql_db_query` . `fbsql_tablename` sert à extraire la liste des tables dans ce résultat.

8.35.45 `fbsql_next_result()` : Déplace le pointeur interne vers le résultat suivant

bool `fbsql_next_result` (resource result_id)

Lorsque vous envoyez plus d'une commande SQL au serveur ou que vous exécutez une procédure stockée avec de multiples résultats, cela va conduire le serveur à retourner plusieurs jeux de lignes. `fbsql_next_result` va vérifier l'existence de plusieurs résultats disponibles sur le serveur. Si un autre jeu de résultats existe, `fbsql_next_result` va détruire le résultat précédent et préparer la lecture dans les nouvelles lignes.

`fbsql_next_result` retourne TRUE si un autre résultat est disponible, ou FALSE sinon.

Exemple avec `fbsql_next_result`

```
<?php
$link = fbsql_connect("localhost", "_SYSTEM", "secret");
fbsql_select_db("MyDB", $link);
$SQL = "Select * from table1; select * from table2;";
$rs = fbsql_query($SQL, $link);
do {
    while ($row = fbsql_fetch_row($rs)) {}
} while (fbsql_next_result($rs));

fbsql_free_result($rs);
fbsql_close($link);
?>
```

8.35.46 `fbsql_num_fields()` : Lit le nombre de champs dans un résultat

int `fbsql_num_fields` (resource result)

`fbsql_num_fields` retourne le nombre de champs dans le résultat result .

Voir aussi `fbsql_db_query` , `fbsql_query` , `fbsql_fetch_field` et `fbsql_num_rows` .

8.35.47 fbsql_num_rows() : Lit le nombre de lignes dans un résultat

int **fbsql_num_rows** (resource result)

fbsql_num_rows retourne le nombre de lignes dans le résultat result . Cette fonction n'est valable qu'avec les commandes SELECT. Pour connaître le nombre de lignes dans une requête INSERT, UPDATE ou DELETE, utilisez **fbsql_affected_rows** .

Exemple fbsql_num_rows

```
<?php
$link = fbsql_connect('localhost', 'utilisateur', 'password');
fbsql_select_db("database", $link);

$result = fbsql_query("SELECT * FROM table1;", $link);
$num_rows = fbsql_num_rows($result);

echo "$num_rows lignes\n";
?>
```

Voir aussi **fbsql_affected_rows** , **fbsql_connect** , **fbsql_select_db** et **fbsql_query** .

8.35.48 fbsql_password() : Lit ou définit le mot de passe à utiliser avec une connexion

string **fbsql_password** (resource link_identifier , string password)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.35.49 fbsql_pconnect() : Ouvre une connexion persistante à un serveur FrontBase

resource **fbsql_pconnect** (string hostname , string username , string password)

fbsql_pconnect retourne une ressource représentant la connexion au serveur FrontBase en cas de succès, ou bien FALSE en cas d'erreur.

fbsql_pconnect établit une connexion persistante à un serveur FrontBase. En cas d'omission, les valeurs suivantes sont utilisées par défaut : host ='localhost', username = nom de l'utilisateur qui possède le processus, et password = pas de mot de passe.

Pour choisir le port d'accès au serveur FrontBase, voyez **fbsql_select_db** .

fbsql_pconnect se comporte comme **fbsql_connect** avec deux différences majeures.

Premièrement, lors de la connexion, **fbsql_pconnect** essaie de trouver une connexion permanente déjà ouverte sur cet hôte, avec les mêmes nom d'utilisateur et mot de passe. Si une telle connexion est trouvée, son identifiant est retourné sans ouvrir de nouvelle connexion.

Deuxièmement, la connexion au serveur fbsql ne sera pas terminée avec la fin du script. Au lieu de cela, le lien sera conservé pour un prochain accès (**fbsql_close** ne terminera pas une connexion persistante établie par **fbsql_pconnect**).

C'est pourquoi ce type de connexion est dit 'persistant'.

8.35.50 `fbsql_query()` : Exécute une requête sur un serveur FrontBase

resource `fbsql_query` (string `query` , resource `link_identifier` , int `batch_size`)

`fbsql_query` envoie la requête `query` à la base de données courante, sur le serveur représenté par sa connexion `link_identifier` . Si `link_identifier` est omis, la dernière connexion ouverte est utilisée. Si aucune connexion n'a été ouverte, `fbsql_query` essaie d'établir une connexion en appelant la fonction `fbsql_connect` sans aucun argument.

Note

La requête doit être terminée par un point-virgule !

`fbsql_query` retourne une ressource en cas de succès, ou FALSE , en cas d'échec. Si TRUE est retourné, cela signifie que la requête est correcte et qu'elle peut être exécutée par le serveur. Cela n'indique rien sur le nombre de lignes concernées ou retournées. Il est tout à fait possible qu'une requête réussisse mais qu'aucune ligne ne soit affectée ni retournée.

La requête suivante est invalide, et `fbsql_query` échouera puis retournera FALSE :

Exemple avec `fbsql_query` (1)

```
<?php
$result = fbsql_query("SELECT * WHERE 1=1;")
    or die("Requête invalide");
?>
```

La requête suivante est invalide si `my_col` n'est pas une colonne dans la table `my_tbl` : `fbsql_query` échouera puis retournera FALSE :

Exemple avec `fbsql_query` (2)

```
<?php
$result = fbsql_query("SELECT my_col FROM my_tbl")
    or die ("Requête invalide");
?>
```

`fbsql_query` échouera si vous n'avez pas les droits d'accès sur l'une des bases de données utilisées dans la requête.

Lorsque la requête réussit, vous pouvez utiliser `fbsql_num_rows` pour savoir combien de lignes ont été retournées par une requête SELECT, ou `fbsql_affected_rows` pour les autres requêtes (DELETE, INSERT, REPLACE et UPDATE).

Pour les requêtes SELECT, `fbsql_query` retourne une ressource de résultat, que vous pouvez passer à `fbsql_result` . Lorsque vous avez fini de lire le résultat, vous pouvez libérer les ressources utilisées en appelant `fbsql_free_result` . Cependant, la mémoire sera automatiquement libérée à la fin du script.

Voir aussi `fbsql_affected_rows` , `fbsql_db_query` , `fbsql_free_result` , `fbsql_result` , `fbsql_select_db` et `fbsql_connect` .

8.35.51 `fbsql_read_blob()` : Lit un BLOB dans une base de données

string `fbsql_read_blob` (string `blob_handle` , resource `link_identifieur`)

`fbsql_read_blob` retourne une chaîne de caractères contenant le BLOB `blob_handle` .

`fbsql_read_blob` lit les données du BLOB dans la base de données. Si une sélection contient une colonne de type BLOB et/ou de type CLOB, FrontBase retournera directement les données lors de la lecture. Ce comportement par défaut peut être modifié avec la fonction `fbsql_set_lob_mode` pour que les fonctions de lecture ne retournent qu'un identifiant de BLOB ou CLOB. Si un identifiant est lu, il faut utiliser la fonction `fbsql_read_blob` pour obtenir la valeur du BLOB.

Exemple avec `fbsql_read_blob`

```
<?php

$link = fbsql_pconnect ("localhost", "_SYSTEM", "secret")
        or die ("Impossible de se connecter");

$sql = "SELECT BLOB_COLUMN FROM BLOB_TABLE;";
$rs = fbsql_query($sql, $link);
$row_data = fbsql_fetch_row($rs);

// $row_data[0] contiendra maintenant les données du blob de la première
// ligne

fbsql_free_result($rs);
$rs = fbsql_query($sql, $link);
fbsql_set_lob_mode($rs, FBSQL_LOB_HANDLE);
$row_data = fbsql_fetch_row($rs);

// $row_data[0] contiendra maintenant un identifiant vers les données du
// blob de la première ligne

$blob_data = fbsql_read_blob($row_data[0]);
fbsql_free_result($rs);
?>
```

Voir aussi `fbsql_create_blob` , `fbsql_read_clob` et `fbsql_set_lob_mode` .

8.35.52 `fbsql_read_clob()` : Lit un CLOB dans une base de données

string `fbsql_read_clob` (string `clob_handle` , resource `link_identifieur`)

`fbsql_read_clob` retourne une chaîne de caractères contenant le CLOB `clob_handle` .

`fbsql_read_clob` lit les données du CLOB dans la base de données. Si une sélection contient une colonne de type CLOB et/ou de type BLOB, FrontBase retournera directement les données lors de la lecture. Ce comportement par défaut peut être modifié avec la fonction `fbsql_set_lob_mode` pour que les fonctions de lecture ne retournent qu'un identifiant de BLOB ou CLOB. Si un identifiant est lu, il faut utiliser la fonction `fbsql_read_clob` pour obtenir la valeur du CLOB.

Exemple avec `fbsql_read_clob`

```
<?php

$link = fbsql_pconnect ("localhost", "_SYSTEM", "secret")
        or die ("Impossible de se connecter");

$sql = "SELECT CLOB_COLUMN FROM CLOB_TABLE;";
```

```

$rs = fbsql_query($sql, $link);
$row_data = fbsql_fetch_row($rs);

// $row_data[0] contiendra maintenant les données du blob de la
// première ligne

fbsql_free_result($rs);
$rs = fbsql_query($sql, $link);
fbsql_set_lob_mode($rs, FBSQL_LOB_HANDLE);
$row_data = fbsql_fetch_row($rs);

// $row_data[0] contiendra maintenant un identifiant vers les
// données du clob de la première ligne

$clob_data = fbsql_read_clob($row_data[0]);
fbsql_free_result($rs);
?>

```

Voir aussi [fbsql_create_blob](#) , [fbsql_read_blob](#) et [fbsql_set_lob_mode](#) .

8.35.53 [fbsql_result\(\)](#) : Lit des données dans un résultat

mixed **fbsql_result** (resource result , int row , mixed field)

[fbsql_result](#) lit le contenu du champ field , dans la ligne row , du résultat result . L'argument field peut être l'index du champ, ou bien son nom, ou bien le nom de sa table plus point plus son nom. Si la colonne a été aliasée, utilisez de préférence l'alias.

Lorsque vous travaillez sur de grands résultats, il est vivement recommandé d'utiliser les fonctions qui lisent toute une ligne d'un coup, plutôt que [fbsql_result](#) qui travaille ligne par ligne. Elles sont beaucoup plus rapides. Notez aussi que les offset numériques sont plus rapides que les offset nominaux.

L'utilisation de [fbsql_result](#) ne doit pas être mélangée avec d'autres fonctions qui utilisent aussi le résultat result .

Alternatives vivement recommandées : [fbsql_fetch_row](#) , [fbsql_fetch_array](#) et [fbsql_fetch_object](#) .

8.35.54 [fbsql_rollback\(\)](#) : Annule une transaction FBSQL

bool **fbsql_rollback** (resource link_identifiant)

[fbsql_rollback](#) termine la transaction courante en annulant toutes les requêtes émises depuis la dernière validation. Cette commande n'est utile que si FrontBase est en mode de validation manuelle.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [fbsql_autocommit](#) et [fbsql_commit](#)

8.35.55 [fbsql_select_db\(\)](#) : Sélectionne une base de données FrontBase

bool **fbsql_select_db** (string database_name , resource link_identifiant)

fbsql_select_db remplace la base de données active courante par database_name , sur la connexion ouverte et représentée par link_identifier . Si link_identifier est omis, la dernière connexion ouverte sera utilisée. Si aucune connexion n'a été ouverte, fbsql_select_db essaiera de se connecter en appelant fbsql_connect sans arguments.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le client contacte FBExec pour connaître le numéro de port à utiliser pour la connexion à la base de données. Si le nom de la base est un numéro, le système l'utilisera comme numéro de port, et ne le demandera pas à FBExec. Le serveur Frontbase peut être démarré avec la commande : FRontBase -FBExec=No -port=<port number> <database name> .

Tous les prochains appels à fbsql_query se feront dans la base database_name .

Si la base de données est protégée avec un mot de passe, l'utilisateur doit appeler la fonction fbsql_database_password avant de sélectionner la base de données.

Voir aussi fbsql_database_password , fbsql_connect , fbsql_pconnect et fbsql_query .

8.35.56 **fbsql_set_lob_mode()** : Modifie le mode de lecture des LOB

bool **fbsql_set_lob_mode** (resource result , string database_name)

fbsql_set_lob_mode retourne TRUE en cas de succès et FALSE sinon.

fbsql_set_lob_mode modifie le mode de lecture des LOB. Lorsque des données BLOB ou CLOB sont stockées dans une base de données FrontBase, elles peuvent l'être de manière directe ou indirecte. Si les données sont stockées directement, elles seront automatiquement lues, quelque soit le mode choisi. Si un LOB fait moins de 512 octets, il sera aussi stocké directement.

- FBSQL_LOB_DIRECT - les données LOB sont lues directement. Lorsque les données sont lues dans la base avec fbsql_fetch_row ou d'autres fonctions de lectures, toutes les colonnes de type CLOB et BLOB seront retournées comme des colonnes habituelles. C'est le comportement par défaut de FrontBase.
- FBSQL_LOB_HANDLE - Les données LOB sont lues sous forme d'identifiants. Lorsque les données sont lues dans la base avec fbsql_fetch_row ou d'autres fonctions de lecture, toutes les colonnes de type CLOB et BLOB seront retournées sous la forme d'un identifiant. Cet identifiant sera une chaîne de 27 caractères de long, formatée comme ceci : "@'00000000000000000000000000000000"

Voir aussi fbsql_create_blob , fbsql_create_clob , fbsql_read_blob et fbsql_read_clob .

8.35.57 **fbsql_set_password()** : Change le mot de passe de l'utilisateur spécifié

bool **fbsql_set_password** (resource link_identifier , string user , string password , string old_password)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.35.58 `fbsql_set_transaction()` : Configure le verrouillage et l'isolation de transactions

void `fbsql_set_transaction` (resource link_identifieur , int Locking , int Isolation)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.35.59 `fbsql_start_db()` : Démarre une base de données FBSQL

bool `fbsql_start_db` (string database_name , resource link_identifieur , string database_options)

`fbsql_start_db` démarre la base de données database_name sur la connexion link_identifieur .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `fbsql_db_status` et `fbsql_stop_db` .

8.35.60 `fbsql_stop_db()` : Stoppe une base de données FBSQL

bool `fbsql_stop_db` (string database_name , resource link_identifieur)

`fbsql_stop_db` stoppe la base de données database_name sur la connexion link_identifieur .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `fbsql_db_status` et `fbsql_start_db` .

8.35.61 `fbsql_tablename()` : Lit le nom de la table d'un champ

string `fbsql_tablename` (resource result , int i)

`fbsql_tablename` retourne le nom de la table d'origine du champ i , dans le résultat result . La fonction `fbsql_num_rows` peut être utilisée pour connaître le nombre de tables dans un résultat.

Exemple avec `fbsql_tablename`

```
<?php
fbsql_connect("localhost", "_SYSTEM", "");
$result = fbsql_list_tables("wisconsin");
$i = 0;
while ($i < fbsql_num_rows($result)) {
    $tb_names[$i] = fbsql_tablename($result, $i);
    echo $tb_names[$i] . "<br />";
    $i++;
}
?>
```

8.35.62 `fbsql_username()` : Lit ou définit le nom de login à utiliser avec une connexion

string `fbsql_username` (resource link_identifieur , string username)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.35.63 `fbsql_warnings()` : Active ou désactive les alertes FrontBase

bool `fbsql_warnings` (bool OnOff)

`fbsql_warnings` retourne TRUE si les alertes sont actives et FALSE en cas d'erreur.

`fbsql_warnings` active ou désactive les alertes FrontBase suivant que onoff est à TRUE ou FALSE .

8.36 Forms Data Format

8.36.1 Introduction

Forms Data Format (FDF) est un format de formulaire pour les documents PDF. Vous pouvez lire la documentation (en anglais) à <http://partners.adobe.com/asn/acrobat/forms.jsp> pour plus de détails sur les tenants et les aboutissants.

L'esprit de FDF est similaire à celui des formulaires HTML. Les différences résident dans les moyens de transmission des données au serveur, lorsque le bouton "submit" (soumettre) est pressé (ce qui est du ressort de Form Data Format) et le format de formulaire lui-même (qui est plutôt du ressort de Portable Document Format, PDF). Gérer des données FDF est un des objectifs des fonctions FDF. Mais il y en a d'autres. Vous pouvez aussi prendre un formulaire PDF, et pré-remplir les champs, sans modifier le formulaire lui-même. Dans ce cas, on va créer un document FDF (`fdf_create`), remplir les champs (`fdf_set_value`) et l'associer à un fichier PDF (`fdf_set_file`). Finalement, le tout sera envoyé au client, avec le type MIME `application/vnd.fdf` . Le module "Acrobat reader" de votre navigateur va reconnaître ce type MIME, et lire le fichier PDF, puis le remplir avec FDF.

Si vous éditez un fichier FDF avec un éditeur de texte, vous trouverez un catalogue d'objet avec le nom de FDF . Cet objet peut contenir des entrées telles que Fields , F , Status etc. Les entrées les plus couramment utilisées sont Fields , qui indique une liste de champs de contrôle, et F qui contient le nom du fichier PDF à qui appartiennent ces données. Ces entrées sont désignées dans la documentation PDF sous le nom de /F-Key ou /Status-Key. La modification de ces entrées est possible avec les fonctions `fdf_set_file` et `fdf_set_status` . Les champs sont modifiables avec les fonctions `fdf_set_value` , `fdf_set_opt` etc.

8.36.2 Pré-requis

Vous aurez besoin du FDF toolkit SDK, disponible sur le site <http://partners.adobe.com/asn/acrobat/forms.jsp> . Depuis PHP 4.3, vous aurez besoin du SDK version 5.0. La bibliothèque FDF toolkit est disponible sous forme de bibliothèque compilée, éditée par Adobe, sur les systèmes d'exploitation Win32, Linux, Solaris et AIX.

8.36.3 Installation

Vous devez compiler PHP avec `--with-fdftk[=DIR]` .

Note

Si vous rencontrez des problèmes lors de la configuration de FDF avec le support `fdftk`, vérifiez que le fichier d'en-tête `fdftk.h` et la bibliothèque `libfdftk.so` sont à leur place. Le fichier de configuration supporte la hiérarchie de dossier de la distribution FDF SDK et l'organisation classique `DIR/include` et `DIR/lib` : vous pouvez donc utiliser l'un ou l'autre directement avec la distribution décompressée, ou bien en incluant le fichier d'en-têtes et la bibliothèque appropriée dans votre système, c'est-à-dire dans `/usr/local/include` et `/usr/local/lib` . Il ne reste plus qu'à configurer avec `--with-fdftk=/usr/local` .

Note

Note aux utilisateurs Win32

Afin d'activer ce module dans l'environnement Windows, vous devez copier les bibliothèques **fdftk.dll** depuis le dossier DLL de PHP/Win32 dans le dossier système SYSTEM32 de votre machine (par exemple : C:\WINNT\SYSTEM32 ou C:\WINDOWS\SYSTEM32).

8.36.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.36.5 Types de ressources

8.36.5.1 fdf

La plupart des fonctions FDF nécessite une ressource de type fdf comme premier argument. Une ressource fdf est une structure qui représente un fichier FDF ouvert. Vous pouvez créer des ressources fdf avec les fonctions fdf_create , fdf_open et fdf_open_string .

8.36.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

- FDFValue (entier)
- FDFStatus (entier)
- FDFFile (entier)
- FDFID (entier)
- FDFf (entier)
- FDFSetFf (entier)
- FDFClearFf (entier)
- FDFFlags (entier)
- FDFSetF (entier)
- FDFClrF (entier)
- FDFAP (entier)
- FDFAS (entier)
- FDFAction (entier)
- FDFAA (entier)
- FDFAPRef (entier)
- FDFIF (entier)
- FDFEnter (entier)
- FDFExit (entier)
- FDFDown (entier)
- FDFUp (entier)
- FDFFormat (entier)
- FDFValidate (entier)
- FDFKeystroke (entier)
- FDFCalculate (entier)
- FDFNormalAP (entier)
- FDFRolloverAP (entier)
- FDFDownAP (entier)

8.36.7 Exemples

Les exemples suivants montrent comment évaluer les données du formulaire.

Evaluer un document FDF

```
<?php
// Ouvrir un fichier FDF depuis une chaîne fournie par l'extension PDF
// Le formulaire PDF contient plusieurs champs texte avec les noms de
// volume, date, comment, publisher, preparer, et two boîtes à cocher
// show_publisher et show_preparer.
$fdf = fdf_open_string($HTTP_FDF_DATA);
$volume = fdf_get_value($fdf, "volume");
echo 'Le champ Volume contient la valeur : "<strong>' . $volume . '</strong>"<br />';

$date = fdf_get_value($fdf, "date");
echo 'La valeur du champ date était "<strong>' . $date . '</strong>"<br />';

$comment = fdf_get_value($fdf, "comment");
echo 'La valeur du champ comment était "<strong>' . $comment . '</strong>"<br />';

if (fdf_get_value($fdf, "show_publisher") == "On") {
    $publisher = fdf_get_value($fdf, "publisher");
    echo "La valeur du champ Publisher était : '<strong>' . $publisher . '</strong><br />";
} else
    echo 'La valeur du champ Publisher ne doit pas être affichée.<br />';

if (fdf_get_value($fdf, "show_preparer") == "On") {
    $preparer = fdf_get_value($fdf, "preparer");
    echo 'La valeur du champ Preparer était "<strong>' . $preparer . '</strong>"<br />';
} else
    echo 'La valeur du champ Preparer ne doit pas être affiché.<br />';
fdf_close($fdf);
?>
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [fdf_add_doc_javascript](#)
- [fdf_add_template](#)
- [fdf_close](#)
- [fdf_create](#)
- [fdf_enum_values](#)
- [fdf_errno](#)
- [fdf_error](#)
- [fdf_get_ap](#)
- [fdf_get_attachment](#)
- [fdf_get_encoding](#)
- [fdf_get_file](#)
- [fdf_get_flags](#)
- [fdf_get_opt](#)
- [fdf_get_status](#)
- [fdf_get_value](#)
- [fdf_get_version](#)
- [fdf_header](#)

- [fdf_next_field_name](#)
- [fdf_open_string](#)
- [fdf_open](#)
- [fdf_remove_item](#)
- [fdf_save_string](#)
- [fdf_save](#)
- [fdf_set_ap](#)
- [fdf_set_encoding](#)
- [fdf_set_file](#)
- [fdf_set_flags](#)
- [fdf_set_javascript_action](#)
- [fdf_set_on_import_javascript](#)
- [fdf_set_opt](#)
- [fdf_set_status](#)
- [fdf_set_submit_form_action](#)
- [fdf_set_target_frame](#)
- [fdf_set_value](#)
- [fdf_set_version](#)

8.36.9 [fdf_add_template\(\)](#) : Ajoute un template dans le document FDF

bool **fdf_add_template** (resource fdfdoc , int newpage , string filename , string template , int rename)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.36.10 [fdf_close\(\)](#) : Ferme un document FDF

void **fdf_close** (resource fdf_document)

[fdf_close](#) ferme le document FDF.

Voir aussi [fdf_open](#) .

8.36.11 [fdf_create\(\)](#) : Crée un nouveau document FDF

resource **fdf_create** (void)

[fdf_create](#) crée un nouveau document FDF. Cette fonction est nécessaire pour ceux qui veulent pré remplir les champs d'un formulaire dans un fichier PDF.

Pré remplir un formulaire PDF
<pre><?php \$outfdf = fdf_create(); fdf_set_value(\$outfdf, "volume", \$volume, 0); fdf_set_file(\$outfdf, "http://testfdf/resultlabel.pdf"); fdf_save(\$outfdf, "outtest.fdf"); fdf_close(\$outfdf); Header("Content-type: application/vnd.fdf"); \$fp = fopen("outtest.fdf", "r"); fpassthru(\$fp);</pre>

```
unlink("outtest.fdf");
?>
```

Voir aussi [fdf_close](#) , [fdf_save](#) et [fdf_open](#) .

8.36.12 `fdf_enum_values()` : Appelle une fonction utilisateur à chaque valeur FDF

bool `fdf_enum_values` (resource `fdfdoc` , callback `function` , *mixed* `userdata`)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.36.13 `fdf_errno()` : Retourne le code d'erreur de la dernière opération FDF

int `fdf_errno` (void)

`fdf_errno` retourne le code d'erreur généré par la dernière opération FDF. Ce code vaut 0 s'il n'y a pas eu d'erreur, ou une valeur non nulle en cas d'échec. Un message d'erreur est accessible via la fonction [fdf_error](#) fonction.

Voir aussi [fdf_error](#) .

8.36.14 `fdf_error()` : Retourne le message d'erreur FDF

string `fdf_error` (int `error_code`)

`fdf_error` retourne le message d'erreur FDF associé au code d'erreur `error_code` . Cette fonction utilise le dernier code d'erreur interne si `error_code` est omis. Sinon, `fdf_error()` est une fonction pratique pour le lire : `fdf_error(fdf_errno())` .

Voir aussi [fdf_errno](#) .

8.36.15 `fdf_get_ap()` : Lit l'apparence d'un champ

bool `fdf_get_ap` (resource `fdf_document` , string `field` , int `face` , string `filename`)

`fdf_get_ap` lit l'apparence du champ `field` (i.e. la valeur de la clé /AP) et le stocke dans le fichier `filename` . Les valeurs possibles de `face` sont `FDFNormalAP` , `FDFRolloverAP` et `FDFDownAP` . L'apparence est stockée dans le fichier `filename` .

8.36.16 `fdf_get_attachment()` : Extrait un fichier intégré dans un document FDF

array `fdf_get_attachment` (resource `fdf_document` , string `fieldname` , string `savepath`)

`fdf_get_attachment` extrait le fichier fieldname téléchargé par le biais du champ "file selection", puis le stocke dans le fichier savepath . savepath peut être le nom d'un fichier ou bien un dossier dans lequel le fichier téléchargé sera créé sous son nom original. Tout fichier déjà existant avec le même nom sera écrasé.

Note

Il semble qu'il n'y ait pas d'autre moyen pour connaître le nom du fichier téléchargé que de le stocker dans un dossier avec savepath puis de lire son nom dans le dossier.

Le tableau retourné contient les champs suivants :

- path - chemin de stockage du dossier
- size - taille du fichier stocké en octets
- type - Type MIMI du fichier, s'il a été fourni dans le document FDF

Enregistrement d'un fichier téléchargé
--

<pre><?php \$fdf = fdf_open_string(\$HTTP_FDF_DATA); \$data = fdf_get_attachment(\$fdf, "filename", "/tmpdir"); echo "Le fichier téléchargé est stocké dans {\$data['path']}"; ?></pre>

8.36.17 `fdf_get_encoding()` : Lit la valeur de la clé /Encoding

string `fdf_get_encoding` (resource fdf_document)

`fdf_get_encoding` retourne la valeur de la clé /Encoding. Une chaîne vide est retournée si le schéma par défaut est utilisé : PDFDocEncoding/Unicode.

Voir aussi `fdf_set_encoding` .

8.36.18 `fdf_get_file()` : Lit la valeur de la clé /F

string `fdf_get_file` (resource fdf_document)

`fdf_set_file` lit la valeur de la clé /F.

Voir aussi `fdf_set_file` .

8.36.19 `fdf_get_flags()` : Lit les attributs d'un champ FDF

int `fdf_get_flags` (resource fdfdoc , string fieldname , int whichflags)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.36.20 `fdf_get_opt()` : Lit une valeur dans un tableau de valeur d'un champ FDF

mixed `fdf_get_opt` (resource `fdfdof` , string `fieldname` , int *element*)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.36.21 `fdf_get_status()` : Lit la valeur de la clé /STATUS

string `fdf_get_status` (resource `fdf_document`)

`fdf_get_status` retourne la valeur de la clé /STATUS.

Voir aussi `fdf_set_status` .

8.36.22 `fdf_get_value()` : Retourne la valeur d'un champ FDF

mixed `fdf_get_value` (resource `fdf_document` , string `fieldname` , int *which*)

`fdf_get_value` retourne la valeur du champ `fieldname` dans le document `fdf_document` .

Les éléments d'un champ tableau peuvent être lus en fournissant le paramètre optionnel `which` , sous la forme d'un entier dont la plus petite valeur sera 0. Pour les champs qui ne sont pas des tableaux, le paramètre optionnel `which` sera ignoré.

Note

Le support des tableaux et du paramètre optionnel <code>which</code> ont été ajoutés en PHP 4.3.
--

Voir aussi `fdf_set_value` .

8.36.23 `fdf_get_version()` : Lit le numéro de version de l'API FDF

string `fdf_get_version` (resource `fdf_document`)

`fdf_get_version` retourne le numéro de version FDF pour le document `fdf_document` , pour l'API si aucun paramètre n'est donné.

Pour la version actuelle du FDF toolkit 5.0, le numéro de version est '5.0' et le numéro de version de document est '1.2', '1.3' ou '1.4'.

Voir aussi `fdf_set_version` .

8.36.24 `fdf_header()` : Emet les en-têtes HTTP spécifiques à FDF

void `fdf_header` (void)

`fdf_header` est une fonction de confort, pour émettre les en-têtes HTTP spécifiques à FDF. Elle permet d'envoyer le type Content-type: avec la valeur `application/vnd.fdf` .

8.36.25 `fdf_next_field_name()` : Lit le nom du champ FDF suivant

string `fdf_next_field_name` (resource `fdf_document` , string `fieldname`)

`fdf_next_field_name` retourne le nom du champ après le champ `fieldname` ou le nom du premier champ, si le second paramètre est NULL .

Détecter tous les noms d'un formulaire FDF
--

<pre><?php \$fdf = fdf_open(\$HTTP_FDF_DATA); for (\$field = fdf_next_field_name(\$fdf); \$field != ""; \$field = fdf_next_field_name(\$fdf, \$field)) { echo "Champ : \$field\n"; } ?></pre>

Voir aussi [fdf_enum_fields](#) et [fdf_get_value](#) .

8.36.26 `fdf_open_string()` : Lit un document FDF à partir d'une chaîne de caractères

resource `fdf_open_string` (string `fdf_data`)

`fdf_open_string` lit les données du formulaire depuis la chaîne `fdf_data` . `fdf_data` doit contenir les données retournées par le formulaire PDF ou créé par les fonctions [fdf_create](#) et [fdf_save_string](#) .

Vous pouvez utiliser `fdf_open_string` avec la variable `$HTTP_FDF_DATA` pour traiter des données de formulaires issues de clients distants.

Accès aux données de formulaire FDF

<pre><?php \$fdf = fdf_open_string(\$HTTP_FDF_DATA); // ... fdf_close(\$fdf); ?></pre>
--

Voir aussi [fdf_open](#) , [fdf_close](#) , [fdf_create](#) et [fdf_save_string](#) .

8.36.27 `fdf_open()` : Ouvre un document FDF

resource `fdf_open` (string `filename`)

`fdf_open` ouvre un fichier avec formulaire. Le fichier doit contenir les données retournées par le formulaire PDF.

Vous pouvez traiter les résultats d'un formulaire PDF en méthode POST en appelant la variable `$HTTP_FDF_DATA` , en l'écrivant dans un fichier et en l'ouvrant en utilisant `fdf_open` . Depuis PHP 4.3, vous pouvez également utilisé `fdf_open_string` qui crée un fichier temporaire et l'efface pour vous.

Accéder aux données du formulaire

<pre><?php</pre>

```
// Sauver le fichier FDF dans un fichier temporaire.
$fdffp = fopen("test.fdf", "w");
fwrite($fdffp, $HTTP_FDF_DATA, strlen($HTTP_FDF_DATA));
fclose($fdffp);

// Ouvrir le fichier temporaire, et utiliser les données.
$fdf = fdf_open("test.fdf");
/* ... */
fdf_close($fdf);
?>
```

Voir aussi [fdf_open_string](#) , [fdf_close](#) , [fdf_create](#) et [fdf_save](#) .

8.36.28 `fdf_remove_item()` : Configure le cadre FDF de destination pour le formulaire

bool `fdf_remove_item` (resource `fdfdoc` , string `fieldname` , int `item`)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.36.29 `fdf_save_string()` : Retourne un document FDF sous forme de chaîne

string `fdf_save_string` (resource `fdf_document`)

`fdf_save_string` retourne le document `fdf_document` sous forme de chaînes de caractères.

Lire un document FDF sous forme de chaîne

```
<?php
$fdf = fdf_create();
fdf_set_value($fdf, "foo", "bar");
$str = fdf_save_string($fdf);
fdf_close($fdf);
echo $str;
?>
```

L'exemple ci-dessus va afficher :

```
%FDF-1.2
%âãĬÓ
1 0 obj
<<
/FDF << /Fields 2 0 R >>
>>
endobj
2 0 obj
[
<< /T (foo)/V (bar)>>
]
endobj
trailer
<<
```

```
/Root 1 0 R
```

```
>>
```

```
%%EOF
```

Voir aussi [fdf_save](#) , [fdf_open_string](#) , [fdf_create](#) et [fdf_close](#) .

8.36.30 [fdf_save\(\)](#) : Sauvegarde un document FDF

```
bool fdf\_save ( resource fdf_document , string filename )
```

[fdf_save](#) sauve un document FDF. Le résultat FDF sera écrit dans le fichier filename . Si le paramètre filename est omis, [fdf_save](#) écrira le FDF sur la sortie standard PHP.

Voir aussi [fdf_save_string](#) , [fdf_create](#) et [fdf_close](#) .

8.36.31 [fdf_set_ap\(\)](#) : Fixe l'apparence d'un champ FDF

```
bool fdf\_set\_ap ( resource fdf_document , string field_name , int face , string filename , int page_number )
```

[fdf_set_ap](#) fixe l'apparence d'un champ (i.e. la valeur de la clé /AP). Les valeurs possibles de face sont sont FDFNormalAP , FDFRolloverAP et FDFDownAP .

8.36.32 [fdf_set_encoding\(\)](#) : Modifie l'encodage des caractères

```
bool fdf\_set\_encoding ( resource fdf_document , string encoding )
```

[fdf_set_encoding](#) modifie l'encodage des caractères du document FDF fdf_document . Le paramètre encoding doit être un nom d'encodage valide. Actuellement, les valeurs supportées sont : " Shift-JIS " , " UHC " , " GBK " ou " BigFive " . Une chaîne de caractères vide remplace la valeur par défaut de l'encodage au schéma PDFDocEncoding/Unicode.

8.36.33 [fdf_set_file\(\)](#) : Crée un document PDF pour y afficher des données FDF

```
bool fdf\_set\_file ( resource fdf_document , string url , string target_frame )
```

[fdf_set_file](#) fixe la valeur de la clé /F. la clé /F est simplement une référence sur un formulaire PDF qui doit être pré-rempli. url doit être fourni sous une forme absolue.

Le cadre utilisé pour l'affichage dans le document peut être sélectionné avec le paramètre optionnel target_frame ou la fonction [fdf_set_target_frame](#) .

```
Passer des données FDF à un second formulaire
```

```
<?php
/* Configure l'en-tête pour Adobe FDF */
fdf_header();

/* Démarre un nouveau fichier FDF */
$fdf = fdf_create();
```

```
/* Assigne au champ "foo" la valeur de "bar" */
fdf_set_value($fdf, "foo", "bar");

/* Indique au client d'afficher les données FDF avec "fdf_form.pdf" */
fdf_set_file($fdf, "http://www.example.com/fdf_form.pdf");

/* Affiche le FDF */
fdf_save();

/* Nettoie */
fdf_close();
?>
```

Voir aussi [fdf_get_file](#) et [fdf_set_target_frame](#) .

8.36.34 [fdf_set_flags\(\)](#) : Modifie une option d'un champ

bool **fdf_set_flags** (resource fdf_document , string fieldname , int whichFlags , int newFlags)

[fdf_set_flags](#) modifie certaines options du champ fieldname .

Voir aussi [fdf_set_opt](#) .

8.36.35 [fdf_set_javascript_action\(\)](#) : Modifie l'action javascript d'un champ

bool **fdf_set_javascript_action** (resource fdf_document , string fieldname , int trigger , string script)

[fdf_set_javascript_action](#) affecte une action javascript au champ fieldname .

Voir aussi [fdf_set_submit_form_action](#) .

8.36.36 [fdf_set_on_import_javascript\(\)](#) : Ajoute du code Javascript à être exécuté lorsque Acrobat ouvre un FDF

bool **fdf_set_on_import_javascript** (resource fdfdoc , string script , bool before_data_import)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Voir aussi [fdf_add_doc_javascript](#) et [fdf_set_javascript_action](#) .

8.36.37 [fdf_set_opt\(\)](#) : Modifie une option d'un champ

bool **fdf_set_opt** (resource fdf_document , string fieldname , int element , string str1 , string str2)

[fdf_set_opt](#) modifie les options du champ fieldname .

Voir aussi [fdf_set_flags](#) .

8.36.38 `fdf_set_status()` : Fixe la valeur de la clé /STATUS

`bool fdf_set_status (resource fdf_document , string status)`

`fdf_set_status` fixe la valeur de la clé /STATUS. Lorsqu'un client reçoit un FDF avec un statut positionné, sa valeur sera présentée dans une boîte d'alerte.

Voir aussi `fdf_get_status` .

8.36.39 `fdf_set_submit_form_action()` : Modifie l'action d'un formulaire

`bool fdf_set_submit_form_action (resource fdf_document , string fieldname , int trigger , string script , int flags)`

`fdf_set_submit_form_action` affecte un javascript au champ `fieldname` , exécuté lors de la validation d'un formulaire.

Voir aussi `fdf_set_javascript_action` .

8.36.40 `fdf_set_target_frame()` : Configure le cadre de destination pour l'affichage du formulaire

`bool fdf_set_target_frame (resource fdf_document , string frame_name)`

`fdf_target_frame` configure le cadre de destination pour l'affichage du résultat PDF défini par `fdf_save_file` .

Voir aussi `fdf_save_file` .

8.36.41 `fdf_set_value()` : Modifie la valeur d'un champ FDF

`bool fdf_set_value (resource fdf_document , string fieldname , mixed value , int isName)`

`fdf_set_value` fixe la valeur du champ `fieldname` . La valeur `value` devra être stocké comme une chaîne de caractères même si c'est un tableau. Dans ce cas, tous les éléments du tableau seront stockés comme un tableau de valeur.

Note

Dans les anciennes versions de la suite FDF, le dernier paramètre déterminait si la valeur du champs devait être convertie en un nom PDF (`isname = 1`) ou positionnée comme une chaîne de caractères (`isname = 0`). La valeur n'est plus dans la suite actuelle, version 5.0. Pour des raisons de compatibilité, cela est toujours supporté comme un paramètre optionnel depuis PHP 4.3, mais ignoré en interne.

Le support des tableaux pour le paramètre `value` a été ajouté depuis PHP 4.3.

Voir aussi `fdf_get_value` et `fdf_remove_item` .

8.36.42 **fdf_set_version()** : Modifie le numéro de version du fichier FDF

bool **fdf_set_version** (resource fdf_document , string version)

fdf_set_version modifie le numéro de version du document fdf_document par la version fdf_document . Certaines fonctionnalités supportées par cette extension ne sont disponibles que pour les nouvelles versions de FDF. Pour le FDF toolkit 5.0 courant, version peut valoir '1.2', '1.3' ou '1.4'.

Voir aussi fdf_get_version .

8.37 Fonctions filePro

8.37.1 Introduction

Ces fonctions permettent de lire des données enregistrées dans des bases de données non modifiables filePro.

filePro est une marque de fP Technologies, Inc. Vous pouvez avoir plus de détails sur filePro en visitant <http://www.fptech.com/> .

8.37.2 Installation

Le support de filePro n'est pas activé par défaut dans PHP. Pour utiliser le support fourni dans la distribution, qui assure l'accès en **lecture-seule** , vous devez compiler PHP avec l'option `--enable-filepro` .

- [Introduction](#)
- [Installation](#)
- [filepro_fieldcount](#)
- [filepro_fieldname](#)
- [filepro_fieldtype](#)
- [filepro_fieldwidth](#)
- [filepro_retrieve](#)
- [filepro_rowcount](#)
- [filepro](#)

8.37.4 filepro_fieldname() : Retourne le nom d'un champ filePro

string **filepro_fieldname** (int field_number)

[filepro_fieldname](#) retourne le nom du champ d'index field_number .

8.37.5 filepro_fieldtype() : Retourne le type d'un champ filePro

string **filepro_fieldtype** (int field_number)

[filepro_fieldtype](#) retourne le type du champ d'index field_number .

8.37.6 filepro_fieldwidth() : Retourne la taille d'un champ filePro

int **filepro_fieldwidth** (int field_number)

[filepro_fieldwidth](#) retourne la taille du champ d'index field_number .

8.37.7 `filepro_retrieve()` : Retourne la valeur d'un champ filePro

string `filepro_retrieve` (int `row_number` , int `field_number`)

`filepro_retrieve` des données à partir de l'endroit spécifié dans la base de données. Le paramètre `row_number` doit être compris entre zéro et le nombre total de lignes - 1 (0.. `filepro_rowcount` - 1). De même, `field_number` accepte des valeurs comprises entre zéro et le nombre de champs - 1 (0.. `filepro_fieldcount` - 1)

Note

Lorsque le <code>safe-mode</code> est activé, PHP vérifie si le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.

8.37.8 `filepro_rowcount()` : Retourne le nombre de lignes dans une base filePro

int `filepro_rowcount` (void)

`filepro_rowcount` retourne le nombre de lignes dans une base filePro.

Note

Lorsque le <code>safe-mode</code> est activé, PHP vérifie si le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.

Voir aussi `filepro` .

8.37.9 `filepro()` : Lit et vérifie un fichier

bool `filepro` (string `directory`)

`filepro` lit et vérifie un fichier, puis enregistre le nombre de champs et de lignes.

Aucun verrouillage n'est pratiqué : il vaut alors mieux ne pas modifier la base filePro lorsqu'elle est ouverte par PHP.

Note

Lorsque le <code>safe-mode</code> est activé, PHP vérifie si le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.

8.38 Système de fichiers

8.38.1 Introduction

8.38.2 Pré-requis

Aucune bibliothèque externe n'est requise pour avoir cette extension, mais si vous voulez le support des grands fichiers LFS sur Linux, vous devez avoir une version récente de glibc et vous devrez compiler PHP avec les options suivantes : `-D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64`.

8.38.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.38.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable	Historique
<code>allow_url_fopen</code>	"1"	PHP_INI_SYSTEM	PHP_INI_ALL dans PHP <= 4.3.4. Disponible depuis PHP 4.0.4.
<code>user_agent</code>	NULL	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
<code>default_socket_timeout</code>	"60"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
<code>from</code>	""	PHP_INI_ALL	
<code>auto_detect_line_endings</code>	"0"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

`allow_url_fopen` booléen

Cette option active les versions étendues des fonctions d'accès aux fichiers, qui savent exploiter les URL. Les extensions par défaut permettent d'accéder aux [fichiers distants](#) avec les protocoles FTP ou HTTP. Certaines extensions comme [zlib](#), peuvent ajouter d'autres fonctionnalités.

Note

Cette option peut seulement être définie dans le `php.ini` pour des raisons de sécurité.

Note

Cette option a été introduite immédiatement après la publication de la version 4.0.3. Pour les versions jusqu'à la, vous ne pouvez désactiver cette fonctionnalité qu'au moment de la compilation, avec l'option `--disable-url-fopen-wrapper`.

Attention

Sous Windows, dans les versions antérieures à la version 4.3, les fonctions suivantes ne supportent pas l'accès aux fichiers distants : `include`, `include_once`,

require , require_once et les fonctions imagecreatefromXXX de l'extension Images .

`user_agent` string

Définit le type d'"user agent" (Définition du navigateur web) utilisé par PHP.

`default_socket_timeout` entier

Durée d'expiration (en secondes) pour les flots basés sur les sockets.

Note

Cette option de configuration a été introduite en PHP 4.3.
--

`from` string

Définit le mot de passe FTP en mode anonyme (en général, votre adresse email).

`auto_detect_line_endings` booléen

Lorsque cette option est activée, PHP va examiner les données lues par fgets et file pour voir si le fichier utilise les conventions de ligne de Unix, MS-Dos ou Macintosh.

Cela permet à PHP de fonctionner avec des systèmes Macintosh, mais par défaut, cette option est désactivée, car cette détection impose un légère pénalité en temps de traitement, mais aussi parce que ceux qui utilisent les retours chariots comme séparateurs auront des soucis de compatibilité.

Note

Cette option de configuration a été introduite en PHP 4.3.
--

8.38.5 Types de ressources

8.38.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

GLOB_BRACE (entier)
 GLOB_ONLYDIR (entier)
 GLOB_MARK (entier)
 GLOB_NOSORT (entier)
 GLOB_NOCHECK (entier)
 GLOB_NOESCAPE (entier)
 PATHINFO_DIRNAME (entier)
 PATHINFO_BASENAME (entier)
 PATHINFO_EXTENSION (entier)
 FILE_USE_INCLUDE_PATH (entier)
 FILE_APPEND (entier)
 FILE_IGNORE_NEW_LINES (entier)
 FILE_SKIP_EMPTY_LINES (entier)

8.38.7 Voir aussi

Pour les fonctions connexes, voyez aussi la section sur les accès aux Dossiers et sur les exécutions de programme .

Pour une liste et une explication sur les différents outils d'accès aux fichiers distants, voyez aussi Liste des protocoles supportés .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Voir aussi](#)
- [basename](#)
- [chgrp](#)
- [chmod](#)
- [chown](#)
- [clearstatcache](#)
- [copy](#)
- [delete](#)
- [dirname](#)
- [disk_free_space](#)
- [disk_total_space](#)
- [diskfreespace](#)
- [fclose](#)
- [feof](#)
- [fflush](#)
- [fgetc](#)
- [fgetcsv](#)
- [fgets](#)
- [fgetss](#)
- [file_exists](#)
- [file_get_contents](#)
- [file_put_contents](#)
- [file](#)
- [fileatime](#)
- [filectime](#)
- [filegroup](#)
- [fileinode](#)
- [filemtime](#)
- [fileowner](#)
- [fileperms](#)
- [filesize](#)
- [filetype](#)
- [flock](#)
- [fnmatch](#)
- [fopen](#)
- [fpassthru](#)
- [fputcsv](#)
- [fputs](#)
- [fread](#)
- [fscanf](#)
- [fseek](#)
- [fstat](#)
- [ftell](#)
- [ftruncate](#)
- [fwrite](#)
- [glob](#)
- [is_dir](#)
- [is_executable](#)

- [is_file](#)
- [is_link](#)
- [is_readable](#)
- [is_uploaded_file](#)
- [is_writable](#)
- [is_writeable](#)
- [link](#)
- [linkinfo](#)
- [lstat](#)
- [mkdir](#)
- [move_uploaded_file](#)
- [parse_ini_file](#)
- [pathinfo](#)
- [pclose](#)
- [popen](#)
- [readfile](#)
- [readlink](#)
- [realpath](#)
- [rename](#)
- [rewind](#)
- [rmdir](#)
- [set_file_buffer](#)
- [stat](#)
- [symlink](#)
- [tempnam](#)
- [tmpfile](#)
- [touch](#)
- [umask](#)
- [unlink](#)

8.38.9 chgrp() : Change le groupe propriétaire d'un fichier

bool **chgrp** (string filename , mixed group)

[chgrp](#) essaie de remplacer le groupe propriétaire courant du fichier filename par group . Seul le super-utilisateur (root) peut changer le groupe propriétaire d'un fichier arbitrairement; les utilisateurs classiques ne peuvent changer le groupe propriétaire d'un fichier que si l'utilisateur propriétaire du fichier est membre du groupe.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note
Cette fonction ne fonctionne pas avec les fichiers distants , car le fichier utilisé doit être accessible sur le système de fichiers local.
Note
Lorsque le safe-mode est activé, PHP vérifie si le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.

Voir aussi [chown](#) et [chmod](#) .

8.38.10 chmod() : Change le mode du fichier

bool **chmod** (string filename , int mode)

chmod remplace le mode du fichier filename par le mode mode .

Il est à noter que le mode mode est considéré comme un nombre en notation octale. Afin de vous en assurer, vous pouvez préfixer cette valeur par un zéro (mode):

Comment utiliser chmod

```

chmod ("/somedir/somefile", 755);
// notation décimale : probablement faux
chmod ("/somedir/somefile", "u+rwx,go+rx");
// chaîne : incorrect
chmod ("/somedir/somefile", 0755);
// notation octale : valeur du mode correcte

```

Le paramètre mode est constitué de trois valeurs octales qui spécifient les droits pour le propriétaire, le groupe du propriétaire et les autres, respectivement. Chaque composant peut être calculé en ajoutant les droits désirés. Le chiffre 1 donne les droits d'exécution, le chiffre 2 les droits d'écriture et le chiffre 4 les droits de lecture. Ajoutez simplement ces nombres pour spécifier les droits voulus. Vous pouvez aussi lire le manuel des systèmes Unix avec man 1 chmod et man 2 chmod .

Comment utiliser le paramètre mode de chmod

```

// Lecture et écriture pour le propriétaire, rien pour les autres
chmod ("/un_dossier/un_fichier", 0600);

// Lecture et écriture pour le propriétaire, lecture pour les autres
chmod ("/un_dossier/un_fichier", 0644);

// Tous pour le propriétaire, lecture et exécution pour les autres
chmod ("/un_dossier/un_fichier", 0755);

// Tout pour le propriétaire, lecture exécution pour le groupe, rien pour les autres
chmod ("/un_dossier/un_fichier", 0750);

```

Note

L'utilisateur courant est l'utilisateur avec lequel PHP fonctionne. Il est probablement différent de l'utilisateur que vous utilisez en mode Shell ou FTP. Le mode ne peut être modifié que par l'utilisateur à qui appartient le fichier sur la plupart des systèmes.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Lorsque le safe mode est activé, PHP vérifie si les fichiers et dossiers que vous allez utiliser ont le même UID (propriétaire) que le script qui est en cours d'exécution. De plus, vous ne pouvez pas modifier les SUID, SGID et sticky bits.

Note

Cette fonction ne fonctionne pas avec les fichiers distants , car le fichier utilisé doit être accessible sur le système de fichiers local.

Voir aussi chown et chgrp .

8.38.11 `chown()` : Change le groupe propriétaire du fichier

bool **chown** (string filename , mixed user)

`chown` change le groupe propriétaire courant du fichier filename en user . Seul le super-utilisateur (root) peut changer arbitrairement le propriétaire d'un fichier.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note
Cette fonction ne fonctionne pas avec les <u>fichiers distants</u> , car le fichier utilisé doit être accessible sur le système de fichiers local.
Note
Lorsque le <u>safe-mode</u> est activé, PHP vérifie si le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.
Voir aussi <u>chmod</u> .

8.38.12 `clearstatcache()` : Efface le cache de stat

void **clearstatcache** (void)

L'appel à la fonction stat ou lstat est relativement coûteux en terme de temps d'exécution. Pour cela, le résultat du dernier appel à l'une des fonctions de statut, (voir la liste ci-dessous), est sauvegardé pour ré-utilisation ultérieure. Si vous voulez forcer la vérification du statut d'un fichier, dans le cas où le fichier aurait pu être modifié ou aurait disparu, vous devez utiliser la fonction clearstatcache afin d'effacer de la mémoire les résultats du dernier appel à la fonction.

Sachez bien que PHP ne met pas en cache les informations concernant un fichier inexistant. Si vous appelez file_exists sur un fichier qui n'existe pas, la fonction retournera FALSE jusqu'à ce que vous créez le fichier. Si vous créez le fichier, la fonction retournera TRUE même si vous effacez le fichier.

Note
Cette fonction met en cache des informations sur les fichiers. Vous n'avez donc besoin d'appeler <u>clearstatcache</u> que si vous faites des opérations multiples sur le dossier, et que vous voulez avoir une version récente des informations.
Les fonctions affectées sont : <u>stat</u> , <u>lstat</u> , <u>file_exists</u> , <u>is_writable</u> , <u>is_readable</u> , <u>is_executable</u> , <u>is_file</u> , <u>is_dir</u> , <u>is_link</u> , <u>filectime</u> , <u>fileatime</u> , <u>filemtime</u> , <u>fileinode</u> , <u>filegroup</u> , <u>fileowner</u> , <u>filesize</u> , <u>filetype</u> , et <u>fileperms</u> .

8.38.13 `copy()` : Copie un fichier

bool **copy** (string source , string dest)

`copy` fait une copie du fichier source vers le fichier dest . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec <u>copy</u>
<pre><?php \$file = 'example.txt'; \$newfile = 'example.txt.bak';</pre>

```

if (!copy($file, $newfile)) {
    echo "La copie du fichier $file n'a pas réussi...\n";
}
?>

```

Si vous souhaitez déplacer un fichier, utilisez la fonction [rename](#) .

Note

Depuis PHP 4.3.0, les deux paramètres source et dest peuvent être des URL si les gestionnaires d'URL ont été activés. Voyez la fonction [fopen](#) pour plus de détails. Si dest est une URL, la copie peut échouer si ce protocole ne supporte pas l'écrasement de fichiers existants.

Attention

Si le fichier de destination dest existe déjà, il sera écrasé.

Note

Compatibilité avec Windows : si vous copiez un fichier avec une taille nulle, [copy](#) retournera FALSE , mais le fichier sera correctement copié.

Voir aussi [move_uploaded_file](#) , [rename](#) et la section du manuel concernant la [gestion des téléchargements de fichiers](#) .

8.38.14 delete() : Effacer

```
void delete ( string file )
```

Ceci est une page d'entrée du manuel pour ceux qui recherchent en fait la fonction [unlink](#) ou [unset](#) .

Voir aussi [unlink](#) et [unset](#) .

8.38.15 dirname() : Renvoie le nom du dossier

```
string dirname ( string path )
```

[dirname](#) retourne le nom du dossier qui contient le fichier ou dossier path .

Sous Windows, les slash (/) et anti-slash (\) sont utilisés comme séparateurs de dossier. Dans les autres environnements, seul le slash (/) est utilisé.

Exemple avec [dirname](#)

```

<?php
    $path = "/etc/passwd";
    $file = dirname ($path);
    // $file vaut "/etc"
?>

```

Note

En PHP 4.0.3, [dirname](#) a été modifié pour être compatible POSIX. Cela signifie notamment que s'il n'y a pas de slash dans le paramètre path , un point ('.') est retourné, indiquant le dossier courant. Sinon, la chaîne retournée est path avec le composant final / supprimé. Notez que cela signifie que vous allez obtenir un slash ou un point là où l'ancien comportement de [dirname](#) retournait une chaîne vide.

[dirname](#) a changé de comportement en PHP 4.3.0. Voyez les exemples suivants :

Différents comportements de `dirname`

```
<?php
//avant PHP 4.3.0
dirname('c:/'); // retourne '.'

//apres PHP 4.3.0
dirname('c:/'); // retourne 'c:'

?>
```

`dirname` a été protégé des données binaires depuis PHP 5.0.0.

Voir aussi `basename` , `pathinfo` et `realpath` .

8.38.16 `disk_free_space()` : Renvoie l'espace disque disponible dans le répertoire

float `disk_free_space` (string directory)

`disk_free_space` retournera le nombre d'octets disponibles sur le disque contenant le dossier `directory` .

Exemple avec `disk_free_space`

```
<?php
$df = disk_free_space("/");
// $df contient le nombre d'octets libres sur "/"
?>
```

Note

Cette fonction ne fonctionne pas avec les fichiers distants , car le fichier utilisé doit être accessible sur le système de fichiers local.

Voir aussi `disk_total_space` .

8.38.17 `disk_total_space()` : Retourne la taille d'un dossier

float `disk_total_space` (string directory)

`disk_total_space` lit récursivement toutes les tailles du dossier `directory` et retourne la somme. `directory` peut être aussi une partition de disque.

Exemple avec `disk_total_space`

```
<?php
$df = disk_total_space("/"); // $df contient le nombre d'octets du dossier "/"
?>
```

Note

Cette fonction ne fonctionne pas avec les fichiers distants , car le fichier utilisé doit être accessible sur le système de fichiers local.

Voir aussi `disk_free_space` .

8.38.18 diskfreespace() : Alias de disk_free_space

Cette fonction est un alias de : disk_free_space .

8.38.19 fclose() : Ferme un fichier

bool **fclose** (resource handle)

fclose ferme le fichier handle .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le pointeur de fichier doit être valide et doit avoir été correctement ouvert par fopen ou fsockopen .

Exemple avec fclose

```

<?php

    $handle = fopen('un_fichier.txt', 'r');

    fclose($handle);

?>

```

8.38.20 feof() : Teste la fin du fichier

bool **feof** (resource handle)

feof retourne TRUE si le pointeur handle est à la fin du fichier ou si une erreur survient, sinon, retourne FALSE .

Attention

Si une connexion ouverte avec fsockopen n'est pas fermée par le serveur, feof attendra qu'un délai limite ne soit atteint pour retourner TRUE . Le délai par défaut vaut 60 secondes. Vous devez utiliser la fonction stream_set_timeout pour modifier cette valeur.

Le pointeur de fichier doit être valide et pointer sur un fichier ouvert avec succès par fopen ou fsockopen (et pas encore fermé par fclose).

Attention

Si le pointeur de fichier passé n'est pas valide, vous obtiendrez une boucle infinie parceque EOF échoue à retourner TRUE.

Exemple avec feof et un pointeur de fichier invalide

```

<?php
// Si le fichier ne peut être lu ou n'existe pas, la fonction fopen retourne FALSE
$file = @fopen("no_such_file", "r");

// FALSE issu de fopen emmetra une alerte et fera quel'on aura une boucle infinie ici
while (!feof($file)) {
}

fclose($file);
?>
?

```

8.38.21 fflush() : Envoie tout le contenu généré dans un fichier

bool **fflush** (resource handle)

fflush force l'écriture de toutes les données bufferisées dans le fichier désigné par handle . fflush retourne TRUE en cas de succès, et FALSE sinon.

handle est un pointeur de fichier ouvert avec fopen , popen , ou fsockopen .

8.38.22 fgetc() : Lit un caractère dans un fichier

string **fgetc** (resource handle)

fgetc retourne une chaîne contenant un seul caractère, lu depuis le fichier pointé par handle . fgetc retourne FALSE à la fin du fichier (contrairement à feof).

Le pointeur de fichier doit être valide, et avoir été correctement ouvert par fopen , popen , ou fsockopen .

Attention

Cette fonction peut retourner FALSE , mais elle peut aussi retourner une valeur équivalent à FALSE , utilisable dans une condition if simple. Utilisez l' opérateur === pour tester la valeur de retour exacte de cette fonction.

Exemple avec fgetc

```
<?php
$fp = fopen('unfichier.txt', 'r');
if (!$fp) {
    echo "Impossible d'ouvrir le fichier unfichier.txt";
}
while (false !== ($char = fgetc($fp))) {
    echo "$char\n";
}
?>
```

Note

Cette fonction gère les chaînes binaires.

Voir aussi fread , fopen , popen , fsockopen et fgets .

8.38.23 fgetcsv() : Renvoie la ligne courante et cherche les champs CSV

array **fgetcsv** (resource handle , int length , string delimiter , string enclosure)

handle

Un pointeur valide sur un fichier ouvert avec fopen , popen ou fsockopen .

length (Optionnel)

Doit être plus grand que la plus grande ligne (en terme de caractères) à lire dans le fichier (y compris le caractère de fin de ligne). Ce paramètre est optionnel depuis PHP 5. Omettre ce paramètre (ou le définir à 0 en PHP 5.0.4 et suivant) fait que la longueur maximale de la ligne n'est pas limitée, ce qui est légèrement plus lent.

delimiter (Optionnel)

Spécifie le séparateur (un seul caractère). Par défaut, c'est la virgule.

enclosure (Optionnel)

Spécifie le caractère de délimitation (un seul caractère). Par défaut, c'est les guillemets doubles. Ce paramètre a été ajouté en PHP 4.3.0.

`fgetcsv` est identique à `fgets` mais `fgetcsv` analyse la ligne qu'il lit et recherche les champs CSV, qu'il va retourner dans un tableau les contenant. Le délimiteur de champs `delimiter` est la virgule, à moins que vous ne fournissiez un troisième argument.

Note

Le paramètre `enclosure` a été ajouté en PHP 4.3.0.

`handle` doit être un pointeur valide, et avoir été correctement ouvert par `fopen`, `popen`, ou `fsockopen`.

`length` doit être plus grand que la plus grande ligne trouvée dans un fichier CSV (en comptant les caractères de fin de ligne). Il est devenu optionnel en PHP 5.

`fgetcsv` retourne FALSE en cas d'erreur, ou en cas de fin du fichier.

Note

Une ligne vide dans un fichier CSV sera retournée sous la forme d'un tableau contenant la valeur NULL et ne sera pas traitée comme une erreur.

Lit et affiche le contenu d'un fichier CSV avec `fgetcsv`

```
<?php
$row = 1;
$handle = fopen("test.csv", "r");
while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
    $num = count($data);
    echo "<p> $num fields in line $row: <br /></p>\n";
    $row++;
    for ($c=0; $c < $num; $c++) {
        echo $data[$c] . "<br />\n";
    }
}
fclose($handle);
?>
```

`fgetcsv` a été protégé des données binaires depuis PHP 4.3.5.

Note

La définition des locales entre en jeu avec cette fonction. Si `LANG` vaut, e.g. `en_US.UTF-8`, les fichiers encodés sur un octet sont mal lus par cette fonction.

Voir aussi `explode`, `file`, `pack` et `fputcsv`.

8.38.24 fgets() : Renvoie la ligne courante sur laquelle se trouve le pointeur du fichier

string `fgets` (resource `handle` , int `length`)

`fgets` retourne la chaîne lue jusqu'à la longueur `length - 1` octet depuis le pointeur de fichier `handle`, ou bien la fin du fichier, ou une nouvelle ligne (qui inclue la valeur retournée), ou encore un EOF (celui qui arrive en premier). Si aucune longueur n'est fournie, la longueur par défaut est de 1 ko ou 1024 octets.

Si une erreur survient, `fgets` retourne FALSE.

8.38.23 fgetcsv() : Renvoie la ligne courante et cherche les champs CSV

Erreur courante :

Les programmeurs habitués à la programmation 'C' noteront que `fgets` ne se comporte pas comme son équivalent C lors de la rencontre de la fin du fichier.

Le pointeur de fichier doit être valide et pointer sur un fichier ouvert avec succès par `fopen` ou `fsockopen` (et pas encore fermé par `fclose`).

Un exemple simple :

Lecture d'un fichier ligne par ligne
<pre><?php \$handle = @fopen("/tmp/inputfile.txt", "r"); if (\$handle) { while (!feof(\$handle)) { \$buffer = fgets(\$handle, 4096); echo \$buffer; } fclose(\$handle); } ?></pre>
Note
Le paramètre <code>length</code> est devenu optionnel depuis PHP 4.2.0. S'il est omis, il prend la valeur par défaut de 1024. Depuis PHP 4.3.0, omettre le paramètre <code>length</code> fait que la fonction va continuer à lire le flux jusqu'à ce que la ligne s'achève. Si la majorité des lignes du fichier dépassent les 8 ko, il est plus efficace pour votre script de spécifier ce paramètre.
Note
<code>fgets</code> est compatible avec les données binaires depuis PHP 4.3. Les versions plus anciennes ne le sont pas.
Note
Si vous avez des problèmes avec PHP qui ne reconnaît pas certaines lignes lors de la lecture de fichier qui ont été créé ou lus sur un Macintosh, vous pouvez activer l'option de configuration <code>auto_detect_line_endings</code> .
Voir aussi <code>fgetss</code> , <code>fread</code> , <code>fgetc</code> , <code>stream_get_line</code> , <code>fopen</code> , <code>popen</code> , <code>fsockopen</code> et <code>stream_set_timeout</code> .

8.38.25 `fgetss()` : Renvoie la ligne courante du fichier et élimine les balises HTML

string `fgetss` (resource `handle` , int `length` , string `allowable_tags`)

`fgetss` est identique à `fgets`, mais `fgetss` supprime toutes les balises HTML et PHP qu'il trouve dans le texte lu. `fgetss` lit jusqu'à `length` octets dans la ressource de fichier `handle`.

Vous pouvez aussi préciser les balises qui seront ignorées dans le troisième paramètre optionnel.

Note
<code>allowable_tags</code> a été ajouté en PHP 3.0.13 et PHP 4.0.0.
Le paramètre <code>length</code> est optionnel depuis PHP 5.
Note
Si vous avez des problèmes avec PHP qui ne reconnaît pas certaines lignes lors de la lecture de

fichier qui ont été créé ou lus sur un Macintosh, vous pouvez activer l'option de configuration `auto_detect_line_endings`.

Voir aussi `fgets`, `fopen`, `fsockopen`, `popen` et `strip_tags`.

8.38.26 `file_exists()` : Vérifie si un fichier existe

bool `file_exists` (string filename)

`file_exists` retourne TRUE si le fichier filename existe, et FALSE sinon.

Sous Windows, utilisez le format de chemin //computernome/share/filename ou \\computernome\share\filename pour vérifier qu'un fichier est disponible sur le partage réseau.

Test d'existence d'un fichier

```
<?php
$filename = '/chemin/jusqu/a/fichier.txt';

if (file_exists($filename)) {
    print "Le fichier $filename existe";
} else {
    print "Le fichier $filename n'existe pas";
}
?>
```

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction `clearstatcache` pour plus de détails.

Array

Attention

Cette fonction retourne FALSE pour les fichiers non accessibles en raison des restrictions du `safe mode`. Cependant, ces fichiers peuvent toujours être `inclus` s'ils sont dans le dossier `safe_mode_include_dir`.

Voir aussi `is_readable`, `is_writable`, `is_file` et `file`.

8.38.27 `file_get_contents()` : Lit tout un fichier dans une chaîne

string `file_get_contents` (string filename , bool use_include_path , resource context , int offset , int maxlen)

`file_get_contents` est identique à la fonction `readfile`, hormis le fait que `file_get_contents` retourne le fichier filename dans une chaîne, à partir de la position offset, et jusqu'à maxlen octets. En cas d'erreur, `file_get_contents` retourne FALSE.

Vous pouvez utiliser l'option `use_include_path` : en la mettant à "1", vous rechercherez aussi dans le dossier `include_path`.

Note

Si vous ouvrez une URI avec des caractères spéciaux, comme des espaces, vous devez encoder cette URI avec la fonction `urlencode`.

Note

Le paramètre context peut être éludé par la valeur NULL.

8.38.28 file_put_contents() : Ecrit une chaîne dans un fichier

int **file_put_contents** (string filename , mixed data , int flags , resource context)

Revient à appeler les fonctions [fopen](#) , [fwrite](#) , et [fclose](#) successivement.

Vous pouvez également spécifier le paramètre data sous forme de tableau (tableau non multi-dimensionnel). C'est l'équivalent à `file_put_contents($filename, join(", $array))` .

Depuis PHP 5.1.0, vous pouvez également passer un flux au paramètre data . Ainsi, le buffer restant de ce flux sera copié dans le fichier spécifié. Ce comportement revient à utiliser la fonction [stream_copy_to_stream](#) .

8.38.29 file() : Lit le fichier et renvoie le résultat dans un tableau

array **file** (string filename , int use_include_path , resource context)

[file](#) est identique à [file_get_contents](#) , hormis le fait que [file](#) retourne le fichier filename dans un tableau. Chaque élément du tableau correspond à une ligne du fichier, et les retour-chariots sont placés en fin de ligne.

Note

Chaque élément du tableau résultat contiendra la nouvelle ligne de fin de chaîne. Il faudra donc utiliser [trim](#) sur cette valeur pour la supprimer.

Vous pouvez utiliser l'option `use_include_path` : en la mettant à "1", vous rechercherez aussi dans le dossier [include_path](#) . [file](#) utilise le contexte context , pour les connexions réseau.

Exemple avec [file](#)

```

<?php
// Lit une page web dans un tableau.
$lines = file ('http://www.example.com/');

// Affiche toutes les lignes du tableau comme code HTML, avec les numéros de ligne
foreach ($lines as $line_num => $line) {
    echo 'Ligne No <strong>' . $line_num . '</strong> : ' . htmlspecialchars($line) . '<br />'. "\n";
}

// Un autre exemple, pour obtenir une page web dans une chaîne. Voir aussi la fonction file_get_contents
$html = implode ('', file ('http://www.example.com/'));
?>

```

Note

Depuis PHP 4.3.0, vous pouvez utiliser [file_get_contents](#) pour lire le contenu d'un fichier dans une chaîne.

En PHP 4.3.0, [file](#) est désormais compatible avec les données binaires.

Note

Le support de contexte a été ajouté en PHP 5.0.0. Pour une description des contextes , référez-vous à [Flux](#) .

Attention

Lorsque vous utilisez SSL, le serveur IIS de Microsoft violera le protocole en fermant la connexion sans envoyer l'indicateur `close_notify`. PHP le reportera en tant que "SSL: Fatal Protocol Error" quand vous arrivez à la fin des données. L'astuce est de baisser le niveau de la directive

[error_reporting](#) pour ne pas inclure les warnings. A partir de PHP 4.3.7, le bogue est détecté automatiquement lors de l'ouverture du flux en utilisant [https://](#) et supprimera cet avertissement pour vous. Si vous utilisez [fsockopen](#) pour créer une socket [ssl://](#), vous devez vous occuper vous-même de supprimer l'erreur.

Voir aussi [readfile](#) , [fopen](#) , [fsockopen](#) , [popen](#) , [file_get_contents](#) et [include](#) .

8.38.30 [fileatime\(\)](#) : Renvoie la date à laquelle le fichier a été accédé pour la dernière fois

int [fileatime](#) (string filename)

[fileatime](#) renvoie la date à laquelle le fichier filename a été accédé pour la dernière fois, ou FALSE en cas d'erreur.

Note : La date de dernière modification d'un fichier est supposé changer à chaque fois que les blocks de données du fichier ont commencés à être lus. Cela peut être très couteux en terme de performance lorsqu'une application accède régulièrement à beaucoup de fichiers ou de répertoires. La plupart des systèmes de fichiers Unix peuvent être montés en ayant désactivés cette information pour accroître les performances d'une telle application ; Les nouvelles USENET sont un bon exemple. Sur de telles systèmes de fichiers, cette fonction devient totalement inutile.

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction [clearstatcache](#) pour plus de détails.

Array

Exemple avec [fileatime](#)

```
<?php
// affiche : un_fichier.txt a été accédé le : December 29 2002 22:16:23.

$filename = 'un_fichier.txt';
if (file_exists($filename)) {
    echo "$filename a été accédé le : " . date("F d Y H:i:s.", fileatime($filename));
}
?>
```

Voir aussi [filemtime](#) , [fileinode](#) et [date](#) .

8.38.31 [filectime\(\)](#) : Renvoie la date de dernier accès à un inode

int [filectime](#) (string filename)

[filectime](#) renvoie l'heure à laquelle l'inode filename a été accédé pour la dernière fois, ou FALSE en cas d'erreur.

Note : Sur la plupart des serveurs UNIX, un fichier est considéré comme modifié si les données de son inode sont modifiées. C'est-à-dire lorsque les permissions (utilisateur, groupe ou autre) ont été modifiées. Voyez aussi [filemtime](#) (que vous pourrez utiliser lorsque vous créerez des indications telles que "Dernière modification : " sur les pages web) et [fileatime](#) .

Notez aussi que sur certains systèmes UNIX, le ctime d'un fichier texte est considéré comme sa date de création. Cela est faux ! Il n'y a pas de date de création de fichier sous la plupart des systèmes UNIX.

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache pour plus de détails.

Array

Exemple avec filectime
--

<pre><?php // affiche e.g. un_fichier.txt a été modifié le : December 29 2002 22:16:23. \$filename = 'somefile.txt'; if (file_exists(\$filename)) { echo "\$filename a été modifié le : " . date("F d Y H:i:s.", filectime(\$filename)); } ?></pre>

Voir aussi [filemtime](#) .

8.38.32 filegroup() : Lire le nom du groupe

int **filegroup** (string filename)

[filegroup](#) renvoie le groupe qui possède le fichier filename , ou FALSE en cas d'erreur. L'identifiant de groupe est retourné au format numérique, utilisez [posix_getgrgid](#) pour retrouver le nom du groupe. En cas d'erreur, FALSE est retourné, et PHP génère un erreur de niveau E_WARNING .

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache pour plus de détails.

Array

Voir aussi [fileowner](#) et [safe_mode_gid](#) .

8.38.33 fileinode() : Renvoie le numéro d'inode du fichier

int **fileinode** (string filename)

[fileinode](#) renvoie le numéro d'inode du fichier filename , ou FALSE en cas d'erreur.

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache pour plus de détails.

Array

Voir aussi [stat](#) .

8.38.34 filemtime() : Renvoie la date de dernière modification du fichier

int **filemtime** (string filename)

filemtime renvoie la date de dernière modification du fichier filename , ou FALSE en cas d'erreur.

filemtime retourne l'heure d'écriture des blocs données d'un fichier. Utilisez date sur ce résultat pour obtenir une date de modification humainement lisible.

Exemple avec filemtime

```
<?php
// l'affichage sera : lefichier.txt a été modifié le : December 29 2002 22:16:23.

$filename = 'lefichier.txt';
if (file_exists($filename)) {
    echo "$filename a été modifié le : " . date ("F d Y H:i:s.", filemtime($filename));
}
?>
```

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache pour plus de détails.

Array

Voir aussi filectime , stat , touch et getlastmod .

8.38.35 fileowner() : Renvoie le nom du propriétaire du fichier

int **fileowner** (string filename)

fileowner renvoie le nom du possesseur du fichier filename , ou FALSE en cas d'erreur.

L'identification du possesseur de fichier est numérique : il faut utiliser posix_getpwuid pour retrouver le nom d'utilisateur.

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache pour plus de détails.

Array

Voir aussi stat .

8.38.36 fileperms() : Renvoie les permissions affectées à un fichier

int **fileperms** (string filename)

fileperms renvoie les permissions affectées au fichier filename , ou FALSE en cas d'erreur.

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache pour plus de détails.

Array

Affichage des permissions en valeur octale

```
<?php
echo substr(sprintf('%o', fileperms('/tmp')), -4);
echo substr(sprintf('%o', fileperms('/etc/passwd')), -4);
?>
```

L'exemple ci-dessus va afficher :

1777

0644

Affichage de toutes les permissions

```
<?php
$perms = fileperms('/etc/passwd');

if (($perms & 0xC000) == 0xC000) {
    // Socket
    $info = 's';
} elseif (($perms & 0xA000) == 0xA000) {
    // Lien symbolique
    $info = 'l';
} elseif (($perms & 0x8000) == 0x8000) {
    // Régulier
    $info = '-';
} elseif (($perms & 0x6000) == 0x6000) {
    // Block spécial
    $info = 'b';
} elseif (($perms & 0x4000) == 0x4000) {
    // Dossier
    $info = 'd';
} elseif (($perms & 0x2000) == 0x2000) {
    // Caractère spécial
    $info = 'c';
} elseif (($perms & 0x1000) == 0x1000) {
    // FIFO pipe
    $info = 'p';
} else {
    // Inconnu
    $info = 'u';
}

// Propriétaire
$info .= (($perms & 0x0100) ? 'r' : '-');
$info .= (($perms & 0x0080) ? 'w' : '-');
$info .= (($perms & 0x0040) ?
    (($perms & 0x0800) ? 's' : 'x' ) :
    (($perms & 0x0800) ? 'S' : '-'));

// Groupe
$info .= (($perms & 0x0020) ? 'r' : '-');
$info .= (($perms & 0x0010) ? 'w' : '-');
$info .= (($perms & 0x0008) ?
    (($perms & 0x0400) ? 's' : 'x' ) :
    (($perms & 0x0400) ? 'S' : '-'));

// Tous
$info .= (($perms & 0x0004) ? 'r' : '-');
$info .= (($perms & 0x0002) ? 'w' : '-');
$info .= (($perms & 0x0001) ?
    (($perms & 0x0200) ? 't' : 'x' ) :
    (($perms & 0x0200) ? 'T' : '-'));

echo $info;
```

```
?>
```

L'exemple ci-dessus va afficher :

```
-r--r--r--
```

Voir aussi [is_readable](#) et [stat](#) .

8.38.37 filesize() : Renvoie la taille d'un fichier

int **filesize** (string filename)

[filesize](#) renvoie la taille du fichier filename , ou FALSE (et génère une erreur de niveau E_WARNING) en cas d'erreur.

Note

Comme le type entier de PHP est signé et que de nombreuses plates-formes utilisent des entiers de 32 bits, [filesize](#) peut retourner des résultats étranges pour les fichiers de taille supérieure à 2 Go. Pour les fichiers entre 2Go et 4 Go, cela peut être contourné avec `sprintf("%u", filesize($file))` .

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction [clearstatcache](#) pour plus de détails.

Array

Exemple avec [filesize](#)

```
<?php
// outputs e.g.  unfichier.txt: 1024 octets

$nom_du_fichier = 'unfichier.txt';
echo $filename . ': ' . filesize($nom_du_fichier) . ' octets';

?>
```

Voir aussi [file_exists](#) .

8.38.38 filetype() : Retourne le type de fichier

string **filetype** (string filename)

[filetype](#) renvoie le type du fichier filename . Les réponses possibles sont : fifo , char , dir , block , link , file et unknown .

[filetype](#) retourne FALSE en cas d'erreur. [filetype](#) va aussi émettre une erreur E_NOTICE si l'appel `stat` échoue, ou si le type de fichier est inconnu.

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction [clearstatcache](#) pour plus de détails.

Array

Exemple avec `filetype`

```
<?php
echo filetype('/etc/passwd'); // fichier
echo filetype('/etc/');      // dossier
?>
```

Voir aussi [is_dir](#) , [is_file](#) , [is_link](#) , [file_exists](#) , [stat](#) et [mime_content_type](#) .

8.38.39 flock() : Verrouille le fichier

bool **flock** (resource handle , int operation , int wouldblock)

PHP dispose d'un système complet de verrouillage de fichiers. Tous les programmes qui accèdent au fichier doivent utiliser la même méthode de verrouillage pour qu'il soit efficace.

Note

[flock](#) est obligatoire sous Windows.

[flock](#) agit sur le fichier handle qui doit avoir été ouvert au préalable. operation est une des valeurs suivantes :

- Acquisition d'un verrou en lecture : operation = LOCK_SH (valeur de 1 pour les versions plus anciennes que PHP 4.0.1).
- Acquisition d'un verrou exclusif en écriture : operation = LOCK_EX (valeur de 2 pour les versions plus anciennes que PHP 4.0.1).
- Libération d'un verrou partagé ou exclusif, operation = LOCK_UN (valeur de 3 pour les versions plus anciennes que PHP 4.0.1).
- Si vous voulez que [flock](#) ne se bloque pas durant le verrouillage, ajoutez LOCK_NB à operation (valeur de 4 pour les versions plus anciennes que PHP 4.0.1).

[flock](#) permet de réaliser un système simple de verrous écriture/lecture, qui peut être utilisé sur n'importe quelle plate-forme (Unix et Windows compris). Le troisième paramètre optionnel &wouldblock vaut par défaut TRUE , si le verrou doit bloquer le script (condition d'erreur EWOULDBLOCK). Le verrou est également levé avec la fonction [fclose](#) (qui est également automatiquement appelée lors de la fin du script).

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec `flock`

```
<?php
$fp = fopen("/tmp/lock.txt", "w+");

if (flock($fp, LOCK_EX)) { // pose un verrou exclusif
    fwrite($fp, "Ecrire dans le fichier\n");
    flock($fp, LOCK_UN); // libère le verrou
} else {
    echo "Impossible de verrouiller le fichier!";
}

fclose($fp);
?>
```

Note

Comme `flock` requiert un pointeur de fichier, vous aurez peut être à utiliser un verrou spécial pour protéger l'accès au fichier que vous voulez tronquer en l'ouvrant en mode d'écriture (avec "w" ou "w+" comme argument de `fopen`).

Attention

`flock` ne fonctionne pas sur NFS ou sur les autres systèmes de fichiers réseaux. Vérifiez la documentation de votre système d'exploitation pour plus de détails.

Sur certains systèmes d'exploitation, `flock` est implémenté au niveau processus. Lorsque vous utilisez une API multi-thread comme ISAPI, vous risquez de ne pas pouvoir avoir confiance en `flock` pour protéger vos fichiers contre d'autres scripts PHP qui fonctionnent en parallèle sur d'autres threads du même serveur.

`flock` n'est pas supporté sur les vieux systèmes de fichiers comme FAT et ses dérivés, et elle retournera forcément FALSE sous ces environnements (ceci est particulièrement vrai pour les utilisateurs de Windows 98).

8.38.40 fnmatch() : Repère un fichier à partir d'un masque

bool **fnmatch** (string pattern , string string , int flags)

`fnmatch` vérifie si la chaîne string va passer le masque Shell pattern . flags est un nom de type de masque.

C'est tout particulièrement pratique pour les noms de fichiers, et il peut être utilisé sur des chaînes de caractères. L'utilisateur moyen de Shell peut être familier avec les masques Shell, ou tout au moins, leurs expressions les plus simples, comme '?' et '*'. De cette façon, utiliser `fnmatch` au lieu de `ereg` ou `preg_match` pour des recherches peut être plus pratique pour les non-initiés.

Vérifier le nom d'une couleur avec un masque Shell

```
<?php
  if(fnmatch("*gr[ae]y", $color)) {
    echo "some form of gray ...";
  }
?>
```

Attention

Actuellement, cette fonction n'est pas disponible pour Windows et les autres systèmes non-POSIX.

Voir aussi `glob` , `ereg` et `preg_match` . Vous pouvez aussi consulter le manuel Unix pour `fnmatch(3)` les valeurs de flags .

8.38.41 fopen() : Ouverture d'un fichier ou d'une URL

resource **fopen** (string filename , string mode , bool use_include_path , resource zcontext)

`fopen` crée une ressource nommée, spécifiée par le paramètre filename , sous la forme d'un flux. Si filename est de la forme "protocole://", filename est supposé être une URL, et PHP va rechercher un gestionnaire de protocole adapté pour lire ce fichier. Si aucun gestionnaire pour ce protocole n'est disponible, PHP va émettre une alerte qui vous permettra de savoir que vous avez des problèmes dans votre script, et il tentera d'exploiter filename comme un fichier classique.

Si PHP décide que le fichier filename est un fichier local, il va essayer d'ouvrir un flux avec ce fichier. Le fichier doit être accessible à PHP. Il vous faut donc vous assurer que vous avez les droits d'accès à ce fichier. Si vous activez le `safe mode` , ou la directive `open_basedir` , d'autres conditions

peuvent aussi s'appliquer.

Si PHP a décidé que filename spécifie un protocole enregistré, et que ce protocole est enregistré comme un protocole réseau, PHP s'assurera que la directive `allow_url_fopen` est activée. Si elle est inactive, PHP va émettre une alerte et l'ouverture va échouer.

Note	
La liste des protocoles supportés est disponible sur Liste des protocoles supportés . Certains protocoles (appelés aussi wrappers ou gestionnaires) supportent des context et/ou des options dans le fichier php.ini . Référez-vous aux pages du manuel traitant le protocole, pour connaître la liste des options qui sont disponibles. (e.g. l'option de php.ini <code>user_agent</code> est utilisée par le gestionnaire http).	
Note	
Le support de contexte a été ajouté en PHP 5.0.0. Pour une description des contextes , référez-vous à Flux .	
Note	
Depuis PHP 4.3.2, le mode par défaut est le mode binaire pour toutes les plates-formes qui font la distinction entre les modes binaire et texte. Si vous rencontrez des problèmes dans vos scripts après une mise à jour, essayez d'utiliser le flag 't' en attendant que vous rendiez votre script plus portable comme mentionné ci-dessous.	
Le paramètre mode spécifie le type d'accès désiré au flux. Il peut prendre les valeurs suivantes :	
mode	Description
'r'	Ouvre en lecture seule, et place le pointeur de fichier au début du fichier.
'r+'	Ouvre en lecture et écriture, et place le pointeur de fichier au début du fichier.
'w'	Ouvre en écriture seule ; place le pointeur de fichier au début du fichier et réduit la taille du fichier à 0. Si le fichier n'existe pas, on tente de le créer.
'w+'	Ouvre en lecture et écriture ; place le pointeur de fichier au début du fichier et réduit la taille du fichier à 0. Si le fichier n'existe pas, on tente de le créer.
'a'	Ouvre en écriture seule ; place le pointeur de fichier à la fin du fichier. Si le fichier n'existe pas, on tente de le créer.
'a+'	Ouvre en lecture et écriture ; place le pointeur de fichier à la fin du fichier. Si le fichier n'existe pas, on tente de le créer.
'x'	Crée et ouvre le fichier en lecture seule ; place le pointeur de fichier au début du fichier. Si le fichier existe déjà, <code>fopen</code> va échouer, en retournant FALSE et en générant une erreur de niveau E_WARNING . Si le fichier n'existe pas, <code>fopen</code> tente de le créer. Ce mode est l'équivalent des options O_EXCL O_CREAT pour l'appel système <code>open(2)</code> sous-jacent. Cette option est supportée à partir de PHP 4.3.2 et fonctionne uniquement avec des fichiers locaux.
'x+'	Crée et ouvre le fichier en lecture et écriture ; place le pointeur de fichier au début du fichier. Si le fichier existe déjà, <code>fopen</code> va échouer, en retournant FALSE et en générant une erreur de niveau E_WARNING . Si le fichier n'existe pas, <code>fopen</code> tente de le créer. Ce mode est l'équivalent des options O_EXCL O_CREAT pour l'appel système <code>open(2)</code> sous-jacent. Cette option est supportée à partir de PHP 4.3.2, et fonctionne uniquement avec des fichiers locaux.
Note	
Les systèmes d'exploitation utilisent différents caractères pour les nouvelles lignes. Lorsque vous écrivez un fichier texte, et insérez une nouvelle ligne, vous devez utiliser le bon caractère pour votre système d'exploitation. Les systèmes Unix utilisent <code>\n</code> comme nouvelle ligne, les systèmes Windows utilisent <code>\r\n</code> , et les systèmes Macintosh utilisent <code>\r</code> .	

Si vous n'utilisez pas le bon caractère de nouvelle ligne lors de l'écriture de vos fichiers, vous risquez d'ouvrir vos fichiers avec des applications qui donneront un aspect 'bizarre' au texte.

Windows propose un mode de traduction ('t'), qui va traduire automatiquement les caractères \n en \r\n lorsque vous travaillez sur le fichier. A l'inverse, vous pouvez utiliser l'option 'b' pour forcer le fichier à être écrit en mode binaire, sans traduction des données. Pour utiliser ces options, ajoutez 'b' ou 't' comme dernier caractère du paramètre mode .

Le mode de traduction par défaut dépend de l'interface SAPI et de la version de PHP que vous utilisez. Nous vous recommandons de toujours spécifier les options de traductions pour des raisons de portabilité. Vous devriez utiliser 't' lorsque vous écrivez des fichiers de texte, et le caractère \n pour définir vos fin de ligne, dans les scripts, mais que vous vous attendez à ce que le fichier soit relu par une application comme Notepad. Vous devriez toujours utiliser l'option 'b' dans les autres cas.

Si vous ne précisez pas 'b' lorsque vous travaillez avec des fichiers binaires, vous pourriez rencontrer des problèmes avec vos données, comme des images corrompues ou des caractères \r\n inopinés.

Note

Pour des raisons de portabilité, il est recommandé de toujours utiliser l'option 'b' lorsque vous ouvrez des fichiers avec `fopen` .

Note

A nouveau, pour des raisons de portabilité, il est fortement recommandé de réécrire les scripts qui utilisent l'option 't' , pour qu'ils utilisent le bon caractère de nouvelle ligne, et le mode 'b' .

Le troisième paramètre optionnel `use_include_path` peut être défini à 1 ou à TRUE pour chercher le fichier dans l' `include_path` .

Si l'ouverture échoue, la fonction retourne FALSE et une alerte E_WARNING sera générée. Vous pouvez utiliser le caractère @ pour supprimer cette alerte.

Exemple avec `fopen`

```
<?php
$handle = fopen("/home/rasmus/file.txt", "r");
$handle = fopen("/home/rasmus/file.gif", "wb");
$handle = fopen("http://www.example.com/", "r");
$handle = fopen("ftp://user:password@example.com/somefile.txt", "w");
?>
```

Si vous rencontrez des problèmes en lecture ou écriture de fichier et que vous utilisez PHP en version module de serveur, n'oubliez pas que les fichiers auxquels vous accédez ne sont pas nécessairement accessibles au processus serveur.

Sous Windows, assurez-vous de bien protéger les anti-slash utilisés dans le chemin du fichier, ou bien utilisez des slash.

Exemple avec `fopen` sous Windows

```
<?php
$handle = fopen("c:\\data\\info.txt", "r");
?>
```

Attention

Lorsque vous utilisez SSL, le serveur IIS de Microsoft violera le protocole en fermant la connexion sans envoyer l'indicateur `close_notify`. PHP le reportera en tant que "SSL: Fatal Protocol Error" quand vous arrivez à la fin des données. L'astuce est de baisser le niveau de la directive

error_reporting pour ne pas inclure les warnings. A partir de PHP 4.3.7, le bogue est détecté automatiquement lors de l'ouverture du flux en utilisant https:// et supprimera cet avertissement pour vous. Si vous utilisez fsockopen pour créer une socket ssl://, vous devez vous occuper vous-même de supprimer l'erreur.

Note

Lorsque le safe-mode est activé, PHP vérifie si le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.

Voir aussi Liste des protocoles supportés , fclose , fgets , fread , fwrite , fsockopen , file , file_exists , is_readable , socket_set_timeout et popen .

8.38.42 fpassthru() : Affiche le reste du fichier

int **fpassthru** (resource handle)

fpassthru lit tout le reste d'un fichier jusqu'à la fin et dirige le résultat vers la sortie standard.

Si une erreur survient, fpassthru retourne FALSE . Sinon, fpassthru retourne le nombre de caractères lus dans handle et passés à la sortie standard.

Le pointeur de fichier doit être valide et pointer sur un fichier ouvert avec succès par fopen ou fsockopen (et pas encore fermé par fclose).

Vous devez appeler la fonction rewind pour réinitialiser le pointeur de fichier au début si vous avez déjà écrit des données dans le fichier.

Si vous voulez uniquement copier le contenu d'un fichier dans le buffer de sortie, sans le modifier auparavant ou placer le pointeur à un endroit particulier, vous devriez utiliser la fonction readfile , ce qui vous évite d'appeler la fonction fopen call.

Note

Lorsque vous utilisez la fonction fpassthru sur un fichier binaire sous Windows, assurez-vous d'avoir ouvert le fichier en mode binaire en ajoutant la lettre b au mode d'accès utilisé dans fopen .

Vous êtes encouragé à utiliser l'option b lorsque vous traitez des fichiers binaires, même si votre système ne le requiert pas, de façon à rendre vos scripts portables.

Utiliser fpassthru avec un fichier binaire

```
<?php
// ouvre un fichier en mode binaire
$name = './img/ok.png';
$fp = fopen($name, 'rb');

// envoie les bons en-têtes
header("Content-Type: image/png");
header("Content-Length: " . filesize($name));

// envoie le contenu du fichier, puis stoppe le script
fpassthru($fp);
exit;
?>
```

Voir aussi readfile , fopen , popen et fsockopen .

8.38.43 fputcsv() : Formate une ligne en CSV et l'écrit dans un fichier

int **fputcsv** (resource handle , array fields , string delimiter , string enclosure)

fputcsv formate la ligne passée sous forme de tableau fields , puis écrit le résultat dans le fichier handle . fputcsv retourne la taille de la chaîne écrite, ou bien FALSE en cas d'échec.

Le paramètre optionnel delimiter spécifie le délimiteur (un seul caractère). Par défaut, c'est la virgule.

Le paramètre enclosure spécifie le caractère d'encadrement (un seul caractère), et par défaut, c'est le guillemet double " .

Exemple avec fputcsv

```
<?php
$list = array (
    'aaa,bbb,ccc,dddd',
    '123,456,789',
    '"aaa","bbb"'
);

$fp = fopen('file.csv', 'w');

foreach ($list as $line) {
    fputcsv($fp, split(',', $line));
}

fclose($fp);
?>
```

Note

Si vous avez des problèmes avec PHP qui ne reconnaît pas certaines lignes lors de la lecture de fichier qui ont été créé ou lus sur un Macintosh, vous pouvez activer l'option de configuration auto_detect_line_endings .

Voir aussi fgetcsv .

8.38.44 fputs() : Alias de fwrite

Cette fonction est un alias de : fwrite .

8.38.45 fread() : Lecture du fichier en mode binaire

string **fread** (resource handle , int length)

fread lit jusqu'à length octets dans le fichier référencé par handle . La lecture s'arrête lorsque length octets ont été lus, ou que l'on a atteint la fin du fichier, ou lorsqu'un paquet devient disponible (le premier des trois).

Retourne la chaîne lue ou FALSE si une erreur survient.

Exemple avec fread

```
<?php
// Lit un fichier, et le place dans une chaîne
```

```
$filename = "/usr/local/something.txt";
$handle = fopen ($filename, "r");
$content = fread ($handle, filesize ($filename));
fclose ($handle);
?>
```

Attention

Sur les systèmes qui différencient les fichiers textes et binaires (i.e. Windows) le fichier doit être ouvert avec la lettre 'b' ajoutée au paramètre de mode de la fonction fopen .

Utilisation des fichiers binaires avec fread

```
<?php
$filename = "c:\\fichiers\\uneimage.gif";
$handle = fopen ($filename, "rb");
$content = fread ($handle, filesize ($filename));
fclose ($handle);
?>
```

Attention

Lorsque vous lisez depuis n'importe quelle source qui n'est pas un fichier local, comme des flux retournés lors de la lecture de fichiers distants ou depuis popen et fsockopen , la lecture s'arrête après la réception d'un paquet. Il faut donc faire des boucles pour collecter les données par paquet, comme présenté ci-dessous.

```
<?php
// Pour PHP 5 et suivant
$handle = fopen("http://www.example.com/", "rb");
$content = stream_get_contents($handle);
fclose($handle);
?>
```

```
<?php
$handle = fopen("http://www.example.com/", "rb");
$content = '';
while (!feof($handle)) {
    $content .= fread($handle, 8192);
}
fclose($handle);
?>
```

Note

Si vous voulez lire le contenu d'un fichier dans une chaîne de caractères, utilisez plutôt file_get_contents qui est bien plus rapide que le code ci-dessus.

Voir aussi fwrite , fopen , fsockopen , popen , fgets , fgetss , file et fpassthru .

8.38.46 fscanf() : Analyse un fichier en fonction d'un format

mixed **fscanf** (resource handle , string format , *mixed* ...)

fscanf est similaire à sscanf , mais elle prend comme entrée un fichier, associé à handle et l'interprète en fonction du format format . Si seulement deux paramètres sont passés à la fonction, les valeurs analysées seront retournées sous forme de tableau. Si des arguments optionnels var1 sont passés, la fonction retournera le nombre de valeurs assignées. Les options doivent être passées par référence.

Tous les caractères blancs de la chaîne de formatage correspondent à autant d'espaces dans le flux d'entrée. Cela signifie qu'une tabulation \t dans la chaîne de format peut remplacer un espace simple dans le flux d'entrée.

Exemple avec <u>fscanf</u>
<pre><?php \$handle = fopen ("users.txt","r"); while (\$userinfo = fscanf (\$handle, "%s\t%s\t%s\n")) { list (\$name, \$profession, \$countrycode) = \$userinfo; //... traitement des données } fclose(\$handle); ?></pre>
Fichier utilisateurs.txt
<pre>javier argonaute pe hiroshi sculpteur jp robert hockeyuer us luigi fleuriste it</pre>
Note
<p>Avant PHP 4.3.0, le nombre maximum de caractères lus dans le fichier était de 512, ou bien jusqu'à la première nouvelle ligne "\n" : en fait, le premier des deux. Depuis PHP 4.3.0, des lignes de n'importe quelle taille peuvent être lues.</p>

Voir aussi [fread](#) , [fgets](#) , [fgetss](#) , [sscanf](#) , [printf](#) et [sprintf](#) .

8.38.47 **fseek()** : Modifie la position du pointeur de fichier

int **fseek** (resource handle , int offset , int whence)

[fseek](#) modifie le curseur de position dans le fichier handle . La nouvelle position mesurée en octets à partir du début du fichier, est obtenue en additionnant la distance offset à la position whence . Ce paramètre peut prendre les valeurs suivantes :

- SEEK_SET - La position finale vaut offset octets.
- SEEK_CUR - La position finale vaut la position courante ajoutée à offset octets.
- SEEK_END - La position finale vaut la position courante par rapport à la fin du fichier, ajoutée de offset .

Si whence n'est pas spécifiée, il vaut par défaut SEEK_SET .

[fseek](#) retourne 0 en cas de succès, et sinon -1. Notez que positionner le pointeur au-delà de la fin du fichier n'est pas une erreur.

Exemple avec <u>fseek</u>
<pre><?php \$fp = fopen('un_fichier.txt'); // lit quelques données \$data = fgets(\$fp, 4096); // retourne au début du fichier // identique à rewind(\$fp); fseek(\$fp, 0);</pre>

```
?>
```

`fseek` ne peut pas être utilisé sur les pointeurs retournés par `fopen` s'ils sont au format HTTP ou FTP. `fseek` donne également des résultats indéfinis pour les flux "append-only" (ouvert avec le flag "a").

Note

Le paramètre whence a été ajouté en PHP 4.0.0.

Note

Si vous ouvrez le fichier avec le mode "a" ou "a+", toutes les données que vous écrirez dans le fichier seront toujours ajoutées, sans se soucier de la position dans le fichier.

Voir aussi `ftell` et `rewind`.

8.38.48 `fstat()` : Lit les informations sur un fichier à partir d'un pointeur de fichier

array `fstat` (resource handle)

`fstat` rassemble les informations sur le fichier dont on connaît le pointeur handle . `fstat` est similaire à la fonction `stat` , hormis le fait qu'elle utilise un pointeur de fichier, au lieu d'un nom de fichier.

Retourne un tableau contenant les statistiques pour le fichier ; le format de ce tableau est décrit en détail sur la page de documentation de la fonction `stat` .

Exemple avec `fstat`

```
<?php
// ouvre un fichier
$fp = fopen("/etc/passwd", "r");

// lit des informations
$fstat = fstat($fp);

// ferme le fichier
fclose($fp);

// affiche le résultat
print_r(array_slice($fstat, 13));

?>
```

L'exemple ci-dessus va afficher quelque chose de similaire à :

```
Array
(
[dev] => 771
[ino] => 488704
[mode] => 33188
[nlink] => 1
[uid] => 0
[gid] => 0
[rdev] => 0
```

```
[size] => 1114
[atime] => 1061067181
[mtime] => 1056136526
[ctime] => 1056136526
[blksize] => 4096
[blocks] => 8
)
```

Note

Cette fonction ne fonctionne pas avec les fichiers distants , car le fichier utilisé doit être accessible sur le système de fichiers local.

8.38.49 ftell() : Renvoie la position du pointeur du fichier

int **ftell** (resource handle)

ftell retourne la position courante du pointeur dans le fichier repéré par le pointeur handle , i.e., son offset.

Si une erreur survient, retourne FALSE .

Le pointeur de fichier doit être valide, et avoir été correctement ouvert par fopen ou popen . ftell donne des résultats non définis pour les flux " append-only " (ouvert avec le flag "a").

Exemple avec ftell

```
<?php
// ouvre un fichier et y lit quelques données
$fp = fopen("/etc/passwd", "r");
$data = fgets($fp, 12);

// Où en sommes-nous ?
echo ftell($fp); // 11

fclose($fp);

?>
```

Voir aussi fopen , popen , fseek et rewind .

8.38.50 ftruncate() : Tronque un fichier

bool **ftruncate** (resource handle , int size)

ftruncate prend le pointeur de fichier handle et le tronque à la taille de size . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Le pointeur du fichier n'est changé qu'en mode **aout** . En mode **écriture** , un appel additionnel à la fonction fseek est nécessaire.

Note

Avant PHP 4.3.3, ftruncate retournait un entier de valeur 1, en cas de succès, au lieu de la valeur boolean TRUE .

Voir aussi fopen et fseek .

8.38.51 fwrite() : Écrit un fichier en mode binaire

int **fwrite** (resource *handle* , string *string* , int *length*)

fwrite écrit le contenu de la chaîne *string* dans le fichier pointé par *handle* . Si la longueur *length* est fournie, l'écriture s'arrêtera après *length* octets, ou à la fin de la chaîne (le premier des deux).

fwrite retourne le nombre d'octets écrits ou FALSE en cas d'erreur.

Notez que si *length* est fourni, alors l'option de configuration `magic_quotes_runtime` sera ignorée, et les slash seront conservés.

Note

Sur les systèmes qui font la différence entre les fichiers binaires et les fichiers textes (par exemple, Windows), le fichier doit être ouvert avec l'option 'b' inclus dans le paramètre de mode de `fopen` .

Exemple simple avec fwrite

```

<?php
$filename = 'test.txt';
$somecontent = "Ajout de chaîne dans le fichier \n";

// Assurons nous que le fichier est accessible en écriture
if (is_writable($filename)) {

    // Dans notre exemple, nous ouvrons le fichier $filename en mode d'ajout
    // Le pointeur de fichier est placé à la fin du fichier
    // c'est là que $somecontent sera placé
    if (!$handle = fopen($filename, 'a')) {
        echo "Impossible d'ouvrir le fichier ($filename)";
        exit;
    }

    // Ecrivons quelque chose dans notre fichier.
    if (fwrite($handle, $somecontent) === FALSE) {
        echo "Impossible d'écrire dans le fichier ($filename)";
        exit;
    }

    echo "L'écriture de ($somecontent) dans le fichier ($filename) a réussi";

    fclose($handle);
} else {
    echo "Le fichier $filename n'est pas accessible en écriture.";
}
?>

```

Voir aussi `fread` , `fopen` , `fsockopen` , `popen` et `file_put_contents` .

8.38.52 glob() : Recherche des chemins qui vérifient un masque

array **glob** (string *pattern* , int *flags*)

glob recherche tous les chemins qui vérifient le masque *pattern* , en suivant les règles utilisées par la fonction `glob()` de la `libc`, qui sont les mêmes que celles utilisées par le Shell en général. Aucun remplacement de tilde (`~`) ou de paramètre n'est fait.

glob retourne un tableau contenant les chemins et fichiers trouvés ou FALSE en cas d'erreur.

Valeurs autorisées pour flags :

- GLOB_MARK Ajoute un slash final à chaque élément retourné
- GLOB_NOSORT : Retourne les fichiers tant l'ordre d'apparence (pas de tri)
- GLOB_NOCHECK : Retourne le masque de recherche si aucun fichier n'a été trouvé
- GLOB_NOESCAPE : Ne protège aucun méta-caractère d'un anti-slash
- GLOB_BRACE : Remplace {a,b,c} par 'a', 'b' ou 'c'
- GLOB_ONLYDIR : Ne retourne que les dossiers qui vérifient le masque

Note

Avant PHP 4.3.3 GLOB_ONLYDIR **n'était pas** disponible sur Windows et les systèmes qui n'utilisent pas la bibliothèque GNU C.

- GLOB_ERR : Stop lors d'une erreur (comme des dossiers non lisibles), par défaut, les erreurs sont ignorées

Note

GLOB_ERR a été ajouté en PHP 5.1.

Un moyen pratique pour remplacer `opendir` par `glob`

```
<?php
$files = glob("*.txt");
foreach ($files as $filename) {
    echo "$filename occupe " . filesize($filename) . " octets\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
funclist.txt occupe 44686 octets
funcsummary.txt occupe 267625 octets
quickref.txt occupe 137820 octets
```

Note

Cette fonction ne fonctionne pas avec les fichiers distants , car le fichier utilisé doit être accessible sur le système de fichiers local.

Voir aussi [opendir](#) , [readdir](#) , [closedir](#) et [fnmatch](#) .

8.38.53 `is_dir()` : Indique si le fichier est un dossier

bool `is_dir` (string filename)

`is_dir` retourne TRUE si filename existe et est un dossier.

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction [clearstatcache](#) pour plus de détails.

Exemple avec `is_dir`

```
<?
var_dump(is_dir('un_fichier.txt')) . "\n";
var_dump(is_dir('bogus_dir/abc')) . "\n";

var_dump(is_dir('../')); //le dossier au-dessus
?>
```

L'exemple ci-dessus va afficher :

```
bool(false)
bool(false)
bool(true)
```

Array

Voir aussi [chdir](#) , [dir](#) , [opendir](#) , [is_file](#) et [is_link](#) .

8.38.54 [is_executable\(\)](#) : Indique si le fichier est exécutable

bool [is_executable](#) (string filename)

[is_executable](#) retourne TRUE si filename existe et est exécutable.

[is_executable](#) est disponible avec Windows depuis PHP 5.0.0.

Exemple avec [is_executable](#)

```
<?php
$file = '/home/vincent/unfichier.sh';

if (is_executable($file)) {
    echo $file.' est exécutable';
} else {
    echo $file.' n'est pas exécutable';
}

?>
```

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction [clearstatcache](#) pour plus de détails.

Array

Voir aussi [is_file](#) et [is_link](#) .

8.38.55 [is_file\(\)](#) : Indique si le fichier est un véritable fichier

bool [is_file](#) (string filename)

[is_file](#) retourne TRUE si filename existe et est un fichier (et pas un dossier).

Exemple avec [is_file](#)

```
<?php
var_dump(is_file('a_file.txt')) . "\n";
var_dump(is_file('/usr/bin/')) . "\n";
?>
```

L'exemple ci-dessus va afficher :

bool(true)
bool(false)

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction [clearstatcache](#) pour plus de détails.

Array

Voir aussi [is_dir](#) et [is_link](#) .

8.38.56 is_link() : Indique si le fichier est un lien symbolique

bool **is_link** (string filename)

[is_link](#) retourne TRUE si filename existe et est un lien symbolique.

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction [clearstatcache](#) pour plus de détails.

Array

Voir aussi [is_dir](#) , [is_file](#) et [readlink](#) .

8.38.57 is_readable() : Indique si un fichier est autorisé en lecture

bool **is_readable** (string filename)

[is_readable](#) retourne TRUE si le fichier ou le dossier spécifié par le paramètre filename existe et est accessible en lecture.

N'oubliez pas que PHP accède aux fichiers avec les mêmes autorisations que l'utilisateur qui fait tourner le serveur web (souvent, c'est 'nobody', personne).

Exemple avec [is_readable](#)

```
<?php
$filename = 'test.txt';
if (is_readable($filename)) {
    echo 'Le fichier est lisible';
} else {
    echo 'Le fichier n\'est pas lisible';
}
?>
```

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction [clearstatcache](#) pour plus de détails.

Array

Voir aussi [is_writable](#) , [file_exists](#) et [fgets](#) .

8.38.58 `is_uploaded_file()` : Indique si le fichier a été téléchargé par HTTP POST

bool `is_uploaded_file` (string filename)

`is_uploaded_file` est disponible à partir des versions PHP 3.0.16 et 4.0.2.

`is_uploaded_file` retourne TRUE si le fichier filename a été téléchargé par HTTP POST. Cela est très utile pour vous assurer qu'un utilisateur n'essaie pas d'accéder intentionnellement à un fichier auquel il n'a pas droit (comme /etc/passwd).

Ce type de vérification est spécialement important s'il est possible que les fichiers téléchargés révèlent leur contenu à l'utilisateur, ou même aux utilisateurs du même système.

Pour un fonctionnement correct, la fonction `is_uploaded_file` nécessite un argument comme `$_FILES['userfile']['tmp_name']`, - le nom du fichier téléchargé sur la machine cliente `$_FILES['userfile']['name']` ne fonctionne pas.

Exemple avec `is_uploaded_file`

```
<?php

if (is_uploaded_file($_FILES['userfile']['tmp_name'])) {
    echo "Fichier ". $_FILES['userfile']['name'] ." téléchargé avec succès.\n";
    echo "Affichage du contenu\n";
    readfile($_FILES['userfile']['tmp_name']);
} else {
    echo "Attaque possible par téléchargement de fichier : ";
    echo "Nom du fichier : '". $_FILES['userfile']['tmp_name'] . "'.";
}

?>
```

`is_uploaded_file` est disponible dans les versions de PHP 3 après PHP 3.0.16, et dans les versions de PHP 4 après la 4.0.2. Si vous êtes coincés avec une version plus ancienne, vous pouvez utiliser l'une des fonctions suivantes pour vous protéger :

Note

L'exemple suivant **ne fonctionne pas** avec les versions de PHP ultérieures à la 4.0.2. Elle dépend de fonctionnalités internes de PHP qui ont disparu après cette version.

Exemple avec `is_uploaded_file` en PHP 4 < 4.0.3.

```
<?php
/* Test utilisateur du fichier téléchargé. */
function is_uploaded_file_4_0_2($filename)
{
    if (!$tmp_file = get_cfg_var('upload_tmp_dir')) {
        $tmp_file = dirname(tempnam('', ''));
    }
    $tmp_file .= '/' . basename($filename);
    /* L'utilisateur peut avoir laissé des / finaux dans php.ini... */
    return (ereg_replace('/+', '/', $tmp_file) == $filename);
}

/* Voici comment l'utiliser, car move_uploaded_file() n'est pas disponible
dans les anciennes versions : */
if (is_uploaded_file_4_0_2($_HTTP_POST_FILES['userfile'])) {
    copy($_HTTP_POST_FILES['userfile'], "/dossier/ou/placer/un/fichier/uploadé");
} else {
    echo "Tentative d'attaque possible : '$_HTTP_POST_FILES[userfile]'. ";
}
?>
```

Voir aussi [move_uploaded_file](#) , et la section sur [la gestion des téléchargements](#) pour un exemple simple.

8.38.59 [is_writable\(\)](#) : Indique si un fichier est autorisé en écriture

bool **is_writable** (string filename)

[is_writable](#) retourne TRUE si filename existe et est accessible en écriture.

N'oubliez pas que PHP accède aux fichiers avec les mêmes autorisations que l'utilisateur qui fait tourner le serveur web (souvent, c'est ' nobody ', personne).

Exemple avec [is_writable](#)

```
<?php
$filename = 'test.txt';
if (is_writable($filename)) {
    echo 'Le fichier est accessible en écriture';
} else {
    echo 'Le fichier n\'est pas accessible en écriture';
}
?>
```

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction [clearstatcache](#) pour plus de détails.

Array

Voir aussi [is_readable](#) , [file_exists](#) et [fwrite](#) .

8.38.60 [is_writeable\(\)](#) : Alias de [is_writable](#)

Cette fonction est un alias de : [is_writable](#) .

8.38.61 [link\(\)](#) : Crée un lien

bool **link** (string target , string link)

[link](#) crée un lien entre le fichier source link et le fichier de destination target . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Cette fonction ne fonctionne pas avec les [fichiers distants](#) , car le fichier utilisé doit être accessible sur le système de fichiers local.

Note

Cette fonction n'est pas implémentée sous Windows.

Voir aussi [symlink](#) pour créer des liens symboliques, [readlink](#) et [linkinfo](#) .

8.38.62 linkinfo() : Renvoie les informations d'un lien

int **linkinfo** (string path)

[linkinfo](#) renvoie les informations du lien path , c'est-à-dire le champ st_dev de la structure C UNIX retournée par lstat . (comme en langage C). [linkinfo](#) sert à vérifier si le lien identifié par path existe, en utilisant la même méthode que la macro S_ISLNK de stat.h . [linkinfo](#) retourne FALSE en cas d'erreur.

Exemple avec [linkinfo](#)

```
<?php
echo linkinfo('/vmlinuz'); // 835
?>
```

Note

Cette fonction n'est pas implémentée sous Windows.

Voir aussi [symlink](#) , [link](#) et [readlink](#) .

8.38.63 lstat() : Retourne les informations sur n fichier ou un lien symbolique

array **lstat** (string filename)

[lstat](#) renvoie les informations sur le fichier filename . Cette fonction est identique à la fonction [stat](#) hormis le fait que si filename est un lien symbolique, les informations seront alors basées sur le lien symbolique.

Voyez la page de manuel de [stat](#) pour plus d'informations sur la structure du tableau retourné par [lstat](#) .

Array

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction [clearstatcache](#) pour plus de détails.

Voir aussi [stat](#) .

8.38.64 mkdir() : Crée un dossier

bool **mkdir** (string pathname , int mode , bool recursive , resource context)

[mkdir](#) tente de créer un dossier dans le chemin pathname .

Notez que vous aurez à préciser le mode en base octale, ce qui signifie que vous aurez probablement un 0 comme premier chiffre. Le mode sera aussi modifié par le umask courant, que vous pouvez modifier avec la fonction [umask](#) .

Note

Le paramètre mode est ignoré sous Windows, et il est devenu optionnel depuis PHP 4.2.0.

Le mode par défaut est le mode 0777, ce qui correspond au maximum de droits possible. Pour plus d'informations sur les modes, lisez en détail la documentation de la fonction [chmod](#) .

Exemple avec [mkdir](#)

```
<?php
mkdir ("/chemin/de/mon/dossier", 0700);
?>
```

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Depuis PHP 5.0.0, la fonction [mkdir](#) peut aussi être utilisée avec **certain**s gestionnaires d'URL. Reportez-vous à [Liste des protocoles supportés](#) , pour une liste des gestionnaires qui supportent [mkdir](#) .

Note

Le support de contexte a été ajouté en PHP 5.0.0. Pour une description des contextes , référez-vous à [Flux](#) .

Note

Le paramètre recursive a été ajouté en PHP 5.0.0.

Note

Lorsque le [safe-mode](#) est activé, PHP vérifie si le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.

Voir aussi [rmdir](#) .

8.38.65 [move_uploaded_file\(\)](#) : Déplace un fichier téléchargé

bool [move_uploaded_file](#) (string filename , string destination)

[move_uploaded_file](#) s'assure que le fichier filename est un fichier téléchargé par HTTP POST. Si le fichier est valide, il est déplacé jusqu'à destination .

Si filename n'est pas valide, rien ne se passe, et [move_uploaded_file](#) retournera FALSE .

Si filename est un fichier téléchargé, mais que pour une raison quelconque, il ne peut être déplacé, rien ne se passe, et [move_uploaded_file](#) retourne FALSE . De plus, une alerte sera affichée.

Ce type de vérification est spécialement important s'il est possible que les fichiers téléchargés révèlent leur contenu à l'utilisateur, ou même aux utilisateurs du même système.

Note

[move_uploaded_file](#) n'est pas affectée par les restrictions liées au [safe mode](#) et à l' [open_basedir](#) . Cependant, les restrictions sont placées uniquement sur le paramètre destination qui permet le déplacement des fichiers chargés dans lesquels filename peut entrer en conflit avec ces restrictions. [move_uploaded_file](#) permet de s'assurer de la sécurité de cette opération en autorisant le déplacement des seuls fichiers chargés via PHP.

Attention

Si le fichier de destination existe déjà, il sera écrasé.

Voir aussi [is_uploaded_file](#) , et la section sur la gestion [des téléchargements de fichiers](#) pour un exemple simple d'utilisation.

8.38.66 parse_ini_file() : Traite un fichier de configuration

array **parse_ini_file** (string filename , bool process_sections)

parse_ini_file charge le fichier filename et retourne les configurations qui s'y trouvent sous forme d'un tableau associatif. En passant le deuxième paramètre optionnel à process_sections , vous obtiendrez un tableau multi-dimensionnel avec les noms des sections. La valeur par défaut de ce paramètre est FALSE

Note

Cette fonction n'a rien à voir avec le fichier php.ini . Ce dernier a déjà été traité lorsque vous commencez à exécuter votre script. Cette fonction peut vous permettre de lire vos propres fichiers de configuration.

La structure des fichiers de configuration lus est similaire à celle de php.ini .

Note

Si une valeur du fichier .ini contient des données non-alphanumériques, il faut la protéger en la plaçant entre guillemets doubles ("").

Note

Depuis PHP 4.2.1, cette fonction est aussi affectée par le safe mode et open_basedir .

Note

Depuis PHP 5.0, cette fonction gère aussi les nouvelles lignes dans les valeurs.

Note

Il y a des mots réservés qui ne doivent pas être utilisés en tant que clés dans les fichiers ini. Cela inclut : null , yes , no , true et false . Les valeurs null , no et false donnent "", yes et true donnent "1". Les caractères |&~![" ne doivent pas être utilisés n'importe où dans la clé et ont une signification spéciale dans la valeur.

La structure du fichier .ini est similaire à celle du fichier php.ini .

Les constantes peuvent aussi être utilisées dans le fichier .ini, ce qui fait que si vous définissez une constante avant d'exécuter parse_ini_file , elle sera intégrée dans les résultats. Seules les valeurs de configuration sont remplacées par leur équivalent en constantes. Par exemple :

Contenu de sample.ini

```
; Ceci est un fichier de configuration
; Les commentaires commencent par ';', comme dans php.ini

[premiere_section]
un = 1
cinq = 5
animal = oiseau

[seconde_section]
chemin = /usr/local/bin
URL = "http://www.example.com/~utilisateur"
```

Exemple avec parse_ini_file

```
<?php

define ('oiseau', 'dodu dodo');

// Analyse sans les sections
$ini_array = parse_ini_file("exemple.ini");
print_r($ini_array);
```

```
// Analyse avec les sections
$ini_array = parse_ini_file("exemple.ini", TRUE);
print_r($ini_array);

?>
```

L'exemple ci-dessus va afficher :

```
Array
(
  [un] => 1
  [cinq] => 5
  [animal] => dodu dodo
  [chemin] => /usr/local/bin
  [URL] => http://www.example.com/~username
)
Array
(
  [premiere_section] => Array
  (
    [un] => 1
    [cinq] => 5
    [animal] => dodu dodo
  )
  [seconde_section] => Array
  (
    [chemin] => /usr/local/bin
    [URL] => http://www.example.com/~utilisateur
  )
)
```

Si les clés et les noms des sections sont des nombres, ils seront évalués comme des entier PHP. Les nombres commençant par 0 sont évalués comme octal et les nombres commençant par 0x sont évalués comme hexadécimal.

8.38.67 pathinfo() : Retourne des informations sur un chemin système

array **pathinfo** (string path , int options)

pathinfo retourne un tableau associatif, contenant les informations sur le chemin path . Les éléments suivants sont retournés : dirname , basename et extension .

Vous pouvez spécifier quels seront les éléments retournés avec le paramètre optionnel options . Il est composé des constantes PATHINFO_DIRNAME , PATHINFO_BASENAME et PATHINFO_EXTENSION . Par défaut, tous les éléments sont retournés.

Exemple avec **pathinfo**

```
<?php
$path_parts = pathinfo('/www/htdocs/index.html');

echo $path_parts['dirname'], "\n";
echo $path_parts['basename'], "\n";
```

```
echo $path_parts['extension'], "\n";
?>
```

L'exemple ci-dessus va afficher :

```
/www/htdocs
index.html
html
```

Note

Pour plus d'informations sur la lecture du chemin courant, lisez la section sur les [variables pré-définies](#) .

Voir aussi [dirname](#) , [basename](#) , [parse_url](#) et [realpath](#) .

8.38.68 pclose() : Ferme un processus de pointeur de fichier

int **pclose** (resource handle)

[pclose](#) ferme le processus de pointeur de fichier handle , ouvert par [popen](#) .

Le pointeur de fichier doit être valide, et avoir été ouvert correctement par [popen](#) .

[pclose](#) retourne le statut final du processus exécuté.

Voir aussi [popen](#) .

8.38.69 popen() : Crée un processus de pointeur de fichier

resource **popen** (string command , string mode)

[popen](#) ouvre un processus fils en faisant un fork de la commande command .

[popen](#) retourne un pointeur de fichier identique à celui retourné par [fopen](#) , hormis le fait qu'il sera unidirectionnel (lecture seule, ou écriture seule), et doit être terminé par [pclose](#) . Ce pointeur peut être utilisé avec [fgets](#) , [fgetss](#) et [fputs](#) .

Si une erreur survient, retourne FALSE .

Exemple avec [popen](#)

```
<?php
    $handle = popen ("/bin/ls", "r");
?>
```

Note

Si la commande à exécuter n'a pu être trouvée, une ressource valide sera retournée. Cela semble étrange, mais c'est pratique. cela vous permet d'accéder aux messages d'erreur qui ont été retournés par le Shell :

Lire des messages d'erreurs avec [popen](#)

```
<?php
    error_reporting(E_ALL);
```

```

/* Ajoute une redirection pour que vous puissiez lire stderr */
$handle = popen('/chemin/vers/spooge 2>&1', 'r');
echo "$handle"; " . gettype($handle) . "\n";
$read = fread($handle, 2096);
echo $read;
pclose($handle);
?>

```

Note

Lorsque le [safe mode](#) est activé, vous pouvez uniquement exécuter des programmes qui se situent dans le dossier défini par [safe_mode_exec_dir](#) . Pour des raisons pratiques, il n'est actuellement pas permis d'avoir le composant .. dans le chemin de l'exécutable.

Attention

Lorsque le [safe mode](#) est activé, la chaîne de commande est échappée avec la fonction [escapeshellcmd](#) . Par exemple, `echo y | echo x` devient `echo y \| echo x` .

Voir aussi [pclose](#) , [fopen](#) et [proc_open](#) .

8.38.70 readfile() : Affiche un fichier

int **readfile** (string filename , bool use_include_path , resource context)

[readfile](#) lit le fichier filename et l'envoie à la sortie standard.

[readfile](#) retourne le nombre d'octets lus depuis le fichier. Si une erreur survient, retourne FALSE , et à moins que la fonction n'ait été appelée avec @readfile, un message d'erreur est affiché.

Array

Vous pouvez utiliser le deuxième paramètre optionnel use_include_path pour explorer le dossier [include_path](#) , en passant la valeur de TRUE .

Vous pouvez utiliser le contexte context pour configurer les accès aux fichiers distants.

Note

Le support de contexte a été ajouté en PHP 5.0.0. Pour une description des contextes , référez-vous à [Flux](#) .

Voir aussi [fpassthru](#) , [file](#) , [fopen](#) , [include](#) , [file_get_contents](#) , [Liste des protocoles supportés](#) , [require](#) et [virtual](#) .

8.38.71 readlink() : Renvoie le nom du fichier vers lequel pointe un lien symbolique

string **readlink** (string path)

[readlink](#) fait la même chose que la fonction readlink en C. Elle retourne le contenu du lien symbolique repéré par path , ou FALSE en cas d'erreur.

Exemple avec [readlink](#)

```

<?php
// affiche par exemple : /boot/vmlinuz-2.4.20-xfs

```

```
echo readlink('/vmlinuz');
?>
```

Note

Cette fonction n'est pas implémentée sous Windows.

Voir aussi [is_link](#) , [symlink](#) et [linkinfo](#) .

8.38.72 realpath() : Retourne le chemin canonique absolu

string **realpath** (string path)

[realpath](#) résout tous les liens symboliques, et remplace toutes les références `./` , `../` et `/` de path puis retourne le chemin canonique absolu ainsi trouvé. Le résultat ne contient aucun lien symbolique, `./` ou `../` .

[realpath](#) retourne FALSE si une erreur survient, e.g. si le fichier n'existe pas. Sous les systèmes BSD, [realpath](#) n'échouera pas s'il manque que le dernier composant du paramètre path , alors que les autres systèmes retourneront FALSE .

Exemple avec [realpath](#)

```
<?php
$real_path = realpath ("../../index.php");
?>
```

Voir aussi [basename](#) , [dirname](#) et [pathinfo](#) .

8.38.73 rename() : Renomme un fichier ou un dossier

bool **rename** (string oldname , string newname , resource context)

[rename](#) tente de renommer oldname en newname .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec [rename](#)

```
<?php
rename("/tmp/tmp_file.txt", "/home/user/login/docs/mon_file.txt");
?>
```

Note

Avant PHP 4.3.3, [rename](#) ne pouvait pas renommer des fichiers entre différentes partitions Unix.

Note

Depuis PHP 5.0.0, [rename](#) peut aussi être utilisée avec **certain**s gestionnaires d'URL. Reportez vous à [Liste des protocoles supportés](#) pour une liste des gestionnaires qui supportent [rename](#) .

Note

Le gestionnaire utilisé dans le paramètre oldname **DOIT** être le même que celui utilisé dans newname .

Note

Le support de contexte a été ajouté en PHP 5.0.0. Pour une description des contexts , référez-vous à [Flux](#) .

Voir aussi [copy](#) , [unlink](#) et [move_uploaded_file](#) .

8.38.74 [rewind\(\)](#) : Remplace le pointeur de fichier au début

bool **rewind** (resource handle)

[rewind](#) remplace le pointeur du fichier handle au début. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le pointeur de fichier doit être valide, et avoir été correctement ouvert par [fopen](#) .

Note
Si vous avez ouvert le fichier en mode d'ajout ("a" ou "a+"), toutes les données que vous écrirez dans ce fichier seront toujours ajoutées, sans se soucier de la position du pointeur de fichier. Voir aussi fseek et ftell .

8.38.75 [rmdir\(\)](#) : Efface un dossier

bool **rmdir** (string dirname , resource context)

[rmdir](#) tente d'effacer le dossier dont le chemin est dirname . Le dossier doit être vide, et le script doit avoir les autorisations adéquates.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note
Depuis PHP 5.0.0, rmdir peut aussi être utilisée avec certain s gestionnaires d'URL. Reportez vous à Liste des protocoles supportés pour une liste des gestionnaires qui supportent rmdir .
Note
Le paramètre context a été ajouté en PHP 5.0.0.
Note
Le support de contexte a été ajouté en PHP 5.0.0. Pour une description des contextes , référez-vous à Flux .
Note
Lorsque le safe-mode est activé, PHP vérifie si le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. Voir aussi mkdir et unlink .

8.38.76 [set_file_buffer\(\)](#) : Alias de [stream_set_write_buffer](#)

Cette fonction est un alias de : [stream_set_write_buffer](#) .

8.38.77 [stat\(\)](#) : Renvoie les informations à propos d'un fichier

array **stat** (string filename)

[stat](#) renvoie les informations à propos du fichier filename . Si filename est un lien symbolique, les informations proviennent du fichier lui-même, et non du lien symbolique. [lstat](#) est identique à [stat](#) sauf que les informations seront alors basées sur le lien symbolique.

En cas d'erreurs, [stat](#) retourne FALSE . Elle générera aussi un avertissement.

Retourne un tableau avec des informations sur le fichier avec les éléments suivants. Ce tableau commence à zéro. En plus de retourner ces attributs dans un tableau indexé, ils peuvent être accédés par leurs clés, comme mentionné à côté de chaque paramètre ; ceci est disponible depuis PHP 4.0.6 :

Numéro	Nom (depuis PHP 4.0.6)	Description
0	dev	volume
1	ino	inode
2	mode	droits d'accès à l'inode
3	nlink	nombre de liens
4	uid	userid du propriétaire
5	gid	groupid du propriétaire
6	rdev	type de volume, si le volume est *
7	size	taille en octets
8	atime	date de dernier accès (Unix timestamp)
9	mtime	date de dernière modification (Unix timestamp)
10	ctime	date de dernier changement d'inode (Unix timestamp)
11	blksize	taille de bloc
12	blocks	nombre de blocs alloués

* - uniquement sur les systèmes qui supportent le type st_blksize . Les autres systèmes (i.e. Windows) retournent -1.

Note
Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache pour plus de détails.

Array

Voir aussi [lstat](#) , [fstat](#) , [filemtime](#) et [filegroup](#) .

8.38.78 symlink() : Crée un lien symbolique

bool **symlink** (string target , string link)

[symlink](#) crée un lien symbolique pour l'objet target avec le nom de link .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note
Cette fonction n'est pas implémentée sous Windows.

Voir aussi [link](#) , [readlink](#) et [linkinfo](#) .

8.38.79 tempnam() : Crée un fichier avec un nom unique

string **tempnam** (string dir , string prefix)

tempnam crée un fichier temporaire unique dans le dossier dir . Si le dossier n'existe pas, tempnam va générer un nom de fichier dans le dossier temporaire du système. Le nom sera préfixé par le paramètre prefix .

Avant PHP 4.0.6, le comportement de tempnam dépendait de l'OS sous-jacent. Sous Windows, la variable d'environnement TMP remplace le paramètre dir ; sous Linux, la variable d'environnement TMPDIR a la priorité, tandis que pour les OS en système V R4, le paramètre dir sera toujours utilisé, si le dossier qu'il représente existe. Consultez votre documentation pour plus de détails.

Note

Si PHP ne peut pas créer un fichier dans le dossier spécifié par le paramètre dir , il tentera dans le dossier par défaut du système.

tempnam retourne le nom du fichier temporaire ou FALSE en cas d'échec.

Exemple avec tempnam

```

<?php
$tmpfname = tempnam ("/tmp", "FOO");

$handle = fopen($tmpfname, "w");
fwrite($handle, "Ecrire dans le fichier temporaire");
fclose($handle);

// faire quelquechose ici

unlink($tmpfname);
?>

```

Note

Le comportement de tempnam a été modifié en PHP 4.0.3. Le fichier temporaire est aussi créé pour éviter des conflits si le fichier apparaît dans le système de fichiers entre le moment où la chaîne est générée et avant que le script n'ait terminé la création du fichier. Notez que vous devez effacer le fichier si vous n'en avez plus besoin, car cela n'est pas fait automatiquement.

Voir aussi tmpfile et unlink .

8.38.80 tmpfile() : Crée un fichier temporaire

resource **tmpfile** (void)

tmpfile crée un fichier temporaire avec un nom unique, ouvert en écriture et lecture (w+), et retourne un pointeur de fichier, identique à ceux retournés par fopen . Ce fichier sera automatiquement effacé lorsqu'il sera fermé (avec fclose), ou lorsque le script sera terminé.

Pour plus de détails, consultez votre documentation système sur la fonction tmpfile(3) , et sur stdio.h .

Exemple avec tmpfile

```

<?php
$tmp = tmpfile();
fwrite($tmp, "Ecriture dans le fichier temporaire");
fseek($tmp, 0);

```

```
echo fread($temp, 1024);
fclose($temp); // Ceci supprime le fichier temporaire
?>
```

L'exemple ci-dessus va afficher :

Écriture dans le fichier temporaire

Voir aussi [tempnam](#) .

8.38.81 touch() : Modifie la date de modification et de dernier accès d'un fichier

bool **touch** (string filename , int time , int atime)

touch tente de forcer la date de modification du fichier désigné par le paramètre filename à la date de spécifiée par le paramètre time . Si time est omis, c'est l'heure courante qui est utilisée. Si le troisième paramètre atime est fourni, il est utilisé comme date de dernier accès. Notez bien que la date de dernier accès est modifiée, quelque soit le nombre de paramètres.

Si le fichier n'existe pas, PHP tente de le créer.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec touch

```
<?php
if ( touch($NomDeFichier) ) {
    echo "La date de modification de $NomDeFichier a
        été fixée à maintenant";
} else {
    echo "Désolé, il est impossible de changer
        la date de modification de $NomDeFichier";
}
?>
```

8.38.82 umask() : Change le "umask" courant

int **umask** (int mask)

umask change le umask de PHP et le remplace par mask : mask & 0777 puis, retourne le vieux umask. Lorsque PHP est utilisé comme module de serveur, le umask reprend sa valeur à la fin de chaque script.

umask , appelée sans arguments, retourne simplement le umask courant.

Note

Évitez d'utiliser cette fonction sur un serveur Web multi-threadé. Il vaut mieux changer les droits d'un dossier avec la fonction [chmod](#) , après la création du dossier. En utilisant umask , vous risquez de rencontrer des comportements indéfinis au niveau des autres scripts et du serveur, car ils utilisent tous le même umask.

Exemple avec umask

```
<?php
$old = umask(0);
chmod("/chemin/un_dossier/un_fichier.txt", 0755);
umask($old);

// vérification
if ($old != umask()) {
    die('An error occured while changing back the umask');
}
?>
```

8.38.83 unlink() : Efface un fichier

bool **unlink** (string filename , *resource context*)

unlink efface filename . Identique à la fonction Unix C unlink() . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Depuis PHP 5.0.0, unlink peut aussi être utilisée avec **quelques** protocoles d'URL. Voyez [Liste des protocoles supportés](#) pour une liste des protocoles supportant unlink .

Note

Le support de contexte a été ajouté en PHP 5.0.0. Pour une description des contexts , référez-vous à [Flux](#) .

Voir aussi [rmdir](#) pour la suppression de dossiers.

8.39 Fonctions FriBiDi

8.39.1 Introduction

FriBiDi est une implémentation libre de [l'algorithme bidirectionnel unicode](#) .

8.39.2 Pré-requis

Vous devez télécharger et installer la bibliothèque [FriBiDi](#) .

8.39.3 Installation

Cette extension [PECL](#) n'est pas intégrée à PHP. Des informations comme les notes sur les nouvelles versions, les téléchargements, les sources des fichiers, les informations concernant les mainteneurs ainsi qu'un CHANGELOG, peuvent être trouvées ici : <http://pecl.php.net/package/fribidi> .

Si vous voulez utiliser ces fonctions, vous devez compiler PHP avec le support Fribidi en utilisant l'option de configuration `--with-fribidi[=DIR]` .

Les utilisateurs de Windows doivent activer la bibliothèque `php_fribidi.dll` dans le `php.ini` pour pouvoir utiliser ces fonctions. Vous pouvez télécharger ces bibliothèques DLL pour ces extensions PECL depuis la page [de téléchargement de PHP](#) sous sur la page <http://snaps.php.net/> .

8.39.4 Configuration à l'exécution

8.39.5 Types de ressources

8.39.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

`FRIBIDI_CHARSET_UTF8` ([entier](#))
`FRIBIDI_CHARSET_8859_6` ([entier](#))
`FRIBIDI_CHARSET_8859_8` ([entier](#))
`FRIBIDI_CHARSET_CP1255` ([entier](#))
`FRIBIDI_CHARSET_CP1256` ([entier](#))
`FRIBIDI_CHARSET_ISIRI_3342` ([entier](#))

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)

- fribidi_log2vis

8.40 Fonctions FTP

8.40.1 Introduction

Les fonctions de cette extension implémentent l'accès au client à un serveur de fichier qui gère le protocole FTP comme défini dans la RFC <http://www.faqs.org/rfcs/rfc959> . Cette extension Cette extension permet à l'accès détaillé sur un serveur FTP fournissant un éventail de commandes pour les scripts. Si vous voulez uniquement lire ou écrire un fichier sur un serveur FTP, utilisez plutôt le gestionnaire [ftp:// wrapper](ftp://wrapper) avec les [fonctions de système de fichiers](#) qui fournissent une interface simple et intuitive.

8.40.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.40.3 Installation

Pour activer le module FTP de votre configuration PHP, il faut utiliser l'option `--enable-ftp` en PHP 4 et supérieur, et l'option `--with-ftp` en PHP 3.

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de bibliothèque supplémentaire pour disposer de ces fonctions.

8.40.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.40.5 Types de ressources

Cette extension utilise un type de ressource qui est un identifiant du lien de connexion FTP, retourné par [ftp_connect](#) ou [ftp_ssl_connect](#) .

8.40.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

FTP_ASCII ([entier](#))
FTP_TEXT ([entier](#))
FTP_BINARY ([entier](#))
FTP_IMAGE ([entier](#))
FTP_TIMEOUT_SEC ([entier](#))
Voir [ftp_set_option](#) pour les détails.

Les constantes suivantes ont été introduites en PHP 4.3.0.

FTP_AUTOSEEK ([entier](#))

Voir [ftp_set_option](#) pour les détails.

FTP_AUTORESUME ([entier](#))

Détermine automatiquement la position de reprise (RESUME) et la position de début pour les requêtes GET et PUT (ne fonctionne qu'avec FTP_AUTOSEEK)

FTP_FAILED ([entier](#))

Le mode asynchrone a échoué

FTP_FINISHED ([entier](#))

Le mode asynchrone a terminé

FTP_MOREDATA ([entier](#))

Le mode asynchrone est encore actif

8.40.7 Exemples

Exemple FTP

```
<?php
// Mise en place d'une connexion basique
$conn_id = ftp_connect($ftp_server);

// Identification avec un nom d'utilisateur et un mot de passe
$login_result = ftp_login($conn_id, $ftp_user_name, $ftp_user_pass);

// Vérification de la connexion
if ((!$conn_id) || (!$login_result)) {
    echo "La connexion FTP a échoué !";
    echo "Tentative de connexion au serveur $ftp_server pour l'utilisateur $ftp_user_name";
    exit;
} else {
    echo "Connexion au serveur $ftp_server, pour l'utilisateur $ftp_user_name";
}

// Chargement d'un fichier
$upload = ftp_put($conn_id, $destination_file, $source_file, FTP_BINARY);

// Vérification du status du chargement
if (!$upload) {
    echo "Le chargement FTP a échoué!";
} else {
    echo "Chargement de $source_file vers $ftp_server en tant que $destination_file";
}

// Fermeture du flux FTP
ftp_close($conn_id);
?>
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [ftp_alloc](#)
- [ftp_cdup](#)
- [ftp_chdir](#)
- [ftp_chmod](#)
- [ftp_close](#)

- [ftp_connect](#)
- [ftp_delete](#)
- [ftp_exec](#)
- [ftp_fget](#)
- [ftp_fput](#)
- [ftp_get_option](#)
- [ftp_get](#)
- [ftp_login](#)
- [ftp_mdtm](#)
- [ftp_mkdir](#)
- [ftp_nb_continue](#)
- [ftp_nb_fget](#)
- [ftp_nb_fput](#)
- [ftp_nb_get](#)
- [ftp_nb_put](#)
- [ftp_nlist](#)
- [ftp_pasv](#)
- [ftp_put](#)
- [ftp_pwd](#)
- [ftp_quit](#)
- [ftp_raw](#)
- [ftp_rawlist](#)
- [ftp_rename](#)
- [ftp_rmdir](#)
- [ftp_set_option](#)
- [ftp_site](#)
- [ftp_size](#)
- [ftp_ssl_connect](#)
- [ftp_systype](#)

8.40.9 [ftp_cdup\(\)](#) : Change de dossier et passe au dossier parent

bool [ftp_cdup](#) (resource ftp_stream)

[ftp_cdup](#) change de dossier et passe au dossier parent.

8.40.10 [ftp_chdir\(\)](#) : Change le dossier FTP courant

bool [ftp_chdir](#) (resource ftp_stream , string directory)

[ftp_chdir](#) change le dossier courant en directory .

8.40.11 [ftp_chmod\(\)](#) : Modifie les droits d'un fichier via FTP

int [ftp_chmod](#) (resource ftp_stream , int mode , string filename)

[ftp_chmod](#) modifie les droits d'accès au fichier filename sur le serveur FTP ftp_stream , en lui attribuant les droits de mode , spécifié sous la forme d'un entier en base octale.

8.40.12 ftp_close() : Ferme une connexion FTP

bool **ftp_close** (resource ftp_stream)

ftp_close ferme la connexion ftp_stream et libère les ressources.

Note

Après avoir appelé cette fonction, vous ne pouvez plus utiliser l'ancienne connexion, et vous devez en ouvrir une nouvelle avec <u>ftp_connect</u> .
--

8.40.13 ftp_connect() : Ouvre une connexion FTP

resource **ftp_connect** (string host , int port , int timeout)

ftp_connect ouvre une connexion FTP avec l'hôte host .

8.40.14 ftp_delete() : Efface un fichier sur un serveur FTP

bool **ftp_delete** (resource ftp_stream , string path)

ftp_delete efface le fichier path sur un serveur FTP.

8.40.15 ftp_exec() : Exécute une commande sur le serveur FTP

bool **ftp_exec** (resource ftp_stream , string command)

ftp_exec envoie une commande SITE EXEC au serveur FTP, pour qu'il exécute le programme command .

8.40.16 ftp_fget() : Télécharge un fichier via FTP dans un fichier local

bool **ftp_fget** (resource ftp_stream , resource handle , string remote_file , int mode , int resumepos)

ftp_fget télécharge le fichier remote_file depuis le serveur FTP et l'écrit dans le fichier identifié par handle .

8.40.17 ftp_fput() : Charge un fichier sur un serveur FTP

bool **ftp_fput** (resource ftp_stream , string remote_file , resource handle , int mode , int startpos)

ftp_fput charge les données issues du fichier identifié par handle jusqu'à la fin du fichier.

8.40.18 `ftp_get_option()` : Lit différentes options pour la connexion FTP courante

mixed `ftp_get_option` (resource ftp_stream , int option)

`ftp_get_option` retourne la valeur de l'option option depuis la connexion FTP spécifiée.

8.40.19 `ftp_get()` : Télécharge un fichier depuis un serveur FTP

bool `ftp_get` (resource ftp_stream , string local_file , string remote_file , int mode , int resumepos)

`ftp_get` télécharge le fichier remote_file depuis le serveur FTP, et le sauve dans le fichier local local_file .

8.40.20 `ftp_login()` : Identification avec une connexion FTP

bool `ftp_login` (resource ftp_stream , string username , string password)

`ftp_login` identifie le flux FTP sur le serveur, avec le nom d'utilisateur username et le mot de passe password .

8.40.21 `ftp_mdtm()` : Retourne la date de dernière modification d'un fichier sur un serveur FTP

int `ftp_mdtm` (resource ftp_stream , string remote_file)

`ftp_mdtm` lit la date de dernière modification d'un fichier distant.

Note
Tous les serveurs ne supportent pas cette fonctionnalité !
Note
<code>ftp_mdtm</code> ne fonctionne pas avec les dossiers.

8.40.22 `ftp_mkdir()` : Crée un dossier sur un serveur FTP

string `ftp_mkdir` (resource ftp_stream , string directory)

`ftp_mkdir` crée le dossier nommé directory sur le serveur FTP.

8.40.23 `ftp_nb_continue()` : Reprend le téléchargement d'un fichier (non bloquant)

int `ftp_nb_continue` (resource ftp_stream)

`ftp_nb_continue` reprend le téléchargement d'un fichier sur la connexion ftp_stream , de manière asynchrone.

8.40.24 ftp_nb_fget() : Lit un fichier sur un serveur FTP, et l'écrit dans un fichier (non bloquant)

int **ftp_nb_fget** (resource ftp_stream , resource handle , string remote_file , int mode , int resumepos)

ftp_nb_fget lit le fichier remote_file présent sur le serveur FTP ftp_stream .

La différence entre cette fonction et ftp_fget est que cette fonction peut lire le fichier de manière asynchrone, afin que votre programme fasse autre chose pendant que le fichier est téléchargé.

8.40.25 ftp_nb_fput() : Ecrit un fichier sur un serveur FTP, et le lit depuis un fichier (non bloquant)

int **ftp_nb_fput** (resource ftp_stream , string remote_file , resource handle , int mode , int startpos)

ftp_nb_fput écrit le fichier remote_file présent sur la machine locale, sur le serveur FTP ftp_stream .

La différence entre cette fonction et ftp_fput est que cette fonction peut lire le fichier de manière asynchrone, afin que votre programme fasse autre chose pendant que le fichier est téléchargé.

8.40.26 ftp_nb_get() : Lit un fichier sur un serveur FTP, et l'écrit dans un fichier (non bloquant)

int **ftp_nb_get** (resource ftp_stream , string local_file , string remote_file , int mode , int resumepos)

ftp_nb_get lit le fichier remote_file présent sur le serveur FTP ftp_stream et le sauvegarde dans un fichier local.

La différence entre cette fonction et ftp_fget est que cette fonction peut lire le fichier de manière asynchrone, afin que votre programme fasse autre chose pendant que le fichier est téléchargé.

8.40.27 ftp_nb_put() : Envoie un fichier sur le serveur FTP (non-bloquant)

int **ftp_nb_put** (resource ftp_stream , string remote_file , string local_file , int mode , int startpos)

ftp_nb_put écrit le fichier remote_file présent sur la machine locale, sur le serveur FTP ftp_stream .

La différence entre cette fonction et ftp_fput est que cette fonction peut lire le fichier de manière asynchrone, afin que votre programme fasse autre chose pendant que le fichier est téléchargé.

8.40.28 ftp_nlist() : Retourne la liste des fichiers dans un dossier

array **ftp_nlist** (resource ftp_stream , string directory)

8.40.29 ftp_pasv() : Active ou désactive le mode passif

bool **ftp_pasv** (resource ftp_stream , bool pasv)

ftp_pasv active ou non le mode passif. En mode passif, les données de connexion sont initiées par le client, plutôt que par le serveur. Ce mode peut être nécessaire lorsque le client est derrière un pare-feu.

8.40.30 ftp_put() : Charge un fichier sur un serveur FTP

bool **ftp_put** (resource ftp_stream , string remote_file , string local_file , int mode , *int startpos*)

ftp_put enregistre le fichier local_file sur le serveur FTP.

8.40.31 ftp_pwd() : Retourne le nom du dossier courant

string **ftp_pwd** (resource ftp_stream)

8.40.32 ftp_quit() : Alias de ftp_close

Cette fonction est un alias de : ftp_close .

8.40.33 ftp_raw() : Envoie une commande FTP brute

array **ftp_raw** (resource ftp_stream , string command)

ftp_raw envoie la commande FTP brute command au serveur FTP identifié par ftp_stream .

8.40.34 ftp_rawlist() : Fait une liste détaillée des fichiers dans un dossier

array **ftp_rawlist** (resource ftp_stream , string directory , *bool recursive*)

ftp_rawlist exécute la commande FTP LIST , et retourne le résultat dans un tableau.

8.40.35 ftp_rename() : Renomme un fichier sur un serveur FTP

bool **ftp_rename** (resource ftp_stream , string oldname , string newname)

ftp_rename renomme le fichier ou le dossier from en to , sur le serveur ftp_stream .

8.40.36 ftp_rmdir() : Efface un dossier FTP

bool **ftp_rmdir** (resource ftp_stream , string directory)

ftp_rmdir efface le dossier directory .

8.40.37 ftp_set_option() : Modifie les options de la connexion FTP

bool **ftp_set_option** (resource ftp_stream , int option , mixed value)

ftp_set_option contrôle diverses options de connexion pour une connexion FTP.

8.40.38 ftp_site() : Envoie la commande SITE au serveur

bool **ftp_site** (resource ftp_stream , string command)

ftp_site envoie la commande SITE au serveur FTP.

Les commandes SITE ne sont pas normalisées, et peuvent varier d'un serveur à l'autre. Elles permettent de gérer notamment les permissions de fichier, et les groupes.

8.40.39 ftp_size() : Retourne la taille d'un fichier

int **ftp_size** (resource ftp_stream , string remote_file)

ftp_size retourne la taille d'un fichier donné en octets.

Note

Tous les serveurs ne supportent pas cette fonctionnalité.

8.40.40 ftp_ssl_connect() : Ouvre une connexion FTP sécurisée avec SSL

resource **ftp_ssl_connect** (string host , int port , int timeout)

ftp_ssl_connect ouvre une connexion FTP sécurisée avec SSL sur l'hôte host spécifié.

Note

Pourquoi cette fonction peut ne pas exister ?

<u>ftp_ssl_connect</u> n'est disponible que si le support <u>OpenSSL</u> est actif dans votre version de PHP. Pour Windows, vous devez compiler votre propre binaire PHP pour qu'il supporte cette fonctionnalité.
--

8.40.41 ftp_systype() : Retourne un identifiant de type de serveur FTP

string **ftp_systype** (resource ftp_stream)

ftp_systype retourne le type de serveur FTP distant.

8.41 Fonctions

8.41.1 Introduction

Ces fonctions effectuent les manipulations liées à la gestion des fonctions.

8.41.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.41.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.41.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.41.5 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Constantes pré-définies](#)
- [call_user_func_array](#)
- [call_user_func](#)
- [create_function](#)
- [func_get_arg](#)
- [func_get_args](#)
- [func_num_args](#)
- [function_exists](#)
- [get_defined_functions](#)
- [register_shutdown_function](#)
- [register_tick_function](#)
- [unregister_tick_function](#)

8.41.7 `call_user_func()` : Appelle une fonction utilisateur

mixed `call_user_func` (*callback function* , *mixed parameter* , *mixed ...*)

`call_user_func` appelle la fonction utilisateur `function_name` , et lui passe les paramètres `parameter`

Exemple avec `call_user_func`

```
<?php
function barbier ($type)
{
    echo "Vous vouliez une coupe $type, pas de problème";
}

call_user_func('barbier', 'à la one again');
call_user_func('barbier', 'militaire');
call_user_func('barbier', 'au bol');
?>
```

Les méthodes d'objets peuvent être appelées statiquement en passant `array($objectname, $methodname)` au paramètre fonction .

Exemple avec `call_user_func` et une méthode de classe

```
<?php
class maclasse {
    function dit_bonjour()
    {
        echo "Bonjour!\n";
    }
}

$classname = "maclasse";

call_user_func(array($classname, 'dit_bonjour'));
?>
```

Note

Notez que les paramètres pour `call_user_func` ne sont pas passés par référence.

```
<?php
function increment (&$var)
{
    $var++;
}

$a = 0;
call_user_func('increment', $a);
echo $a; // 0

call_user_func_array('increment', array(&$a)); // Vous pouvez utiliser ceci à la place
echo $a; // 1
?>
```

Voir aussi [is_callable](#) , [call_user_func_array](#) et information à propos de [callback](#) type.

8.41.8 `create_function()` : Crée une fonction anonyme (style lambda)

string **create_function** (string args , string code)

`create_function` crée une fonction anonyme, à partir des paramètres passés, et retourne un nom de fonction unique. Généralement, les arguments `args` sont présentés sous la forme d'une chaîne à guillemets simples, et la même recommandation vaut pour `code` . La raison de l'utilisation des guillemets simples est de protéger les noms de variables du remplacement par leur valeur. Si vous utilisez les guillemets doubles, n'oubliez pas d'échapper les noms de variables (i.e. `\$avar`).

Vous pouvez utiliser cette fonction pour (par exemple) créer une fonction à partir d'informations récoltées durant l'exécution.

Création d'une fonction anonyme avec `create_function`

```
<?php
$newfunc = create_function('$a,$b', 'return "ln($a) + ln($b) = " . log($a * $b);');
echo "Nouvelle fonction anonyme : $newfunc\n";
echo $newfunc(2,M_E).\n";
// affichera :
// Nouvelle fonction anonyme : lambda_1
// ln(2) + ln(2.718281828459) = 1.6931471805599
?>
```

Ou, pour pouvoir appliquer une fonction générique à une liste d'arguments.

Traitement générique par fonction avec `create_function`

```
<?php
function process($var1, $var2, $farr)
{
    foreach ($farr as $f) {
        echo $f($var1, $var2) . "\n";
    }
}

// Création d'une série de fonction mathématiques
$f1 = 'if ($a >=0) {return "b*a^2 = ".$b*sqrt($a);} else {return FALSE;}';
$f2 = "return \"min(b^2+a, a^2,b) = \".min(\$a*\$a+\$b, \$b*\$b+\$a);";
$f3 = 'if ($a > 0 && $b != 0) {return "ln(a)/b = ".log($a)/$b; } else {return FALSE;}';
$farr = array(
    create_function('$x,$y', 'return "un peu de trigo : ".(sin($x) + $x*cos($y));'),
    create_function('$x,$y', 'return "une hypoténuse: ".sqrt($x*$x + $y*$y);'),
    create_function('$a,$b', $f1),
    create_function('$a,$b', $f2),
    create_function('$a,$b', $f3)
);

echo "\nUtilisation de la première liste de fonctions anonymes\n";
echo "paramètres: 2.3445, M_PI\n";
process(2.3445, M_PI, $farr);

// Maintenant une liste de fonction sur chaîne de caractères
$garr = array(
    create_function('$b,$a','if (strcmp($a,$b,3) == 0) return "*** \"$a\" \".
    'et \"$b\""\n** Ces chaînes de ressemblent!! (regarde les trois premiers caractères);'),
    create_function('$a,$b','; return "CRCs: ".crc32($a)." , ".crc32(b);'),
    create_function('$a,$b','; return "similarité(a,b) = ".similar_text($a,$b,&$p).("($p%)"');')
);
echo "\nUtilisation de la seconde liste de fonctions anonymes\n";
process("Twas brillling and the slithy toves", "Twas the night", $garr);
?>
```

L'exemple ci-dessus va afficher :

Utilisation de la première liste de fonctions anonymes

paramètres: 2.3445, M_PI

un peu de trigo : -1.6291725057799

une hypoténuse: 3.9199852871011

$b*a^2 = 4.8103313314525$

$\min(b^2+a, a^2,b) = 8.6382729035898$

$\ln(a/b) = 0.27122299212594$

Utilisation de la seconde liste de fonctions anonymes

** "Twas the night" et "Twas brillling and the slithy toves"

** Ces chaînes se ressemblent ! (regardez les trois premiers caractères)

CRCs: -725381282 , 1908338681

similarité(a,b) = 11(45.833333333333%)

Mais l'utilisation la plus courante des fonctions lambda est la fonction de callback, par exemple, lors de l'utilisation de [array_walk](#) ou [usort](#)

Utilisation de fonctions anonymes comme fonction de callback

```
<?php
$av = array("la ", "une ", "cette ", "une certaine ");
array_walk($av, create_function('&$v,$k','$v = $v."maison";'));
print_r($av);
?>
```

L'exemple ci-dessus va afficher :

Array

```
(
[0] => la maison
[1] => une maison
[2] => cette maison
[3] => une certaine maison
)
```

un tableau de chaînes de caractères ordonnées de la plus courte à la plus longue

```
<?php
$sv = array("petite", "longue", "une très longue chaîne", "une phrase toute entière");
print_r($sv);
?>
```

L'exemple ci-dessus va afficher :

Array

```
(
[0] => petite
[1] => longue
[2] => une très longue chaîne
[3] => une phrase toute entière
)
```

ordonnées de la plus longue à la plus courte

```
<?php
usort($sv, create_function('$a,$b','return strlen($b) - strlen($a);'));
print_r($sv);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[0] => une phrase toute entière
[1] => une très longue chaîne
[2] => longue
[3] => petite
)
```

8.41.9 `func_get_arg()` : Retourne un élément de la liste des arguments

mixed `func_get_arg` (int `arg_num`)

`func_get_arg` retourne l'argument à la position `arg_num` dans la liste d'arguments d'une fonction utilisateur. Les arguments sont comptés en commençant à zéro. `func_get_arg` générera une alerte si elle est appelée hors d'une fonction. Cette fonction ne peut être directement utilisée en tant que paramètre d'une fonction. A la place, son résultat peut être assigné à une variable, qui peut être passée à une fonction.

Si `arg_num` est plus grand que le nombre d'arguments passés, une alerte est générée et la fonction retourne FALSE .

Note

Parce que cette fonction dépend de la portée courante pour déterminer les détails des paramètres, ils ne peuvent être utilisés en tant que paramètre d'une fonction. Si vous devez passer cette valeur, assignez les résultats à une variable et utilisez-la.

Exemple avec `func_get_arg`

```
<?php
function foo()
{
    $numargs = func_num_args();
    echo "Nombre d'arguments: $numargs<br />\n";
    if ($numargs >= 2) {
        echo "Le second argument est: " . func_get_arg (1) . "<br />\n";
    }
}

foo(1, 2, 3);
?>
```

`func_get_arg` peut être utilisé conjointement à `func_num_args` et `func_get_args` pour permettre aux fonctions utilisateurs d'accepter un nombre variable d'arguments.

8.41.10 `func_get_args()` : Retourne les arguments d'une fonction sous forme de tableau

array `func_get_args` (void)

`func_get_args` retourne un tableau dont chaque élément est une copie du membre correspondant de la liste d'arguments de la fonction. `func_get_args` générera une alerte si elle est appelée hors d'une fonction. Cette fonction ne peut être directement utilisée en tant que paramètre d'une

fonction. A la place, son résultat peut être assigné à une variable, qui peut être passée à une fonction.

Note

Cette fonction retourne uniquement une copie des arguments passés, et ne compte pas en tant qu'arguments par défaut (non passés).

Note

Parce que cette fonction dépend de la portée courante pour déterminer les détails des paramètres, ils ne peuvent être utilisés en tant que paramètre d'une fonction. Si vous devez passer cette valeur, assignez les résultats à une variable et utilisez-la.

Exemple avec `func_get_args`

```
<?php
function foo()
{
    $numargs = func_num_args();
    echo "Nombre d'arguments: $numargs<br />\n";
    if ($numargs >= 2) {
        echo "Le second argument est : " . func_get_arg (1) . "<br />\n";
    }
    $arg_list = func_get_args();
    for ($i = 0; $i < $numargs; $i++) {
        echo "L'argument $i est : " . $arg_list[$i] . "<br />\n";
    }
}

foo(1, 2, 3);
?>
```

`func_get_arg` peut être utilisé conjointement à `func_num_args` et `func_get_args` pour permettre aux fonctions utilisateurs d'accepter un nombre variable d'arguments.

8.41.11 `func_num_args()` : Retourne le nombre d'arguments passés à la fonction

int `func_num_args` (void)

`func_num_args` retourne le nombre d'arguments passés à la fonction utilisateur courante.

`func_num_args` générera une alerte si elle est appelée hors d'une fonction. Cette fonction ne peut être directement utilisée en tant que paramètre d'une fonction. A la place, son résultat peut être assigné à une variable, qui peut être passée à une fonction.

Note

Parce que cette fonction dépend de la portée courante pour déterminer les détails des paramètres, ils ne peuvent être utilisés en tant que paramètre d'une fonction. Si vous devez passer cette valeur, assignez les résultats à une variable et utilisez-la.

Exemple avec `func_num_args`

```
<?php
function foo()
{
    $numargs = func_num_args();
    echo "Nombre d'arguments: $numargs\n";
}

foo (1, 2, 3); // affiche 'Nombre d'arguments: 3'
```

```
?>
```

`func_get_arg` peut être utilisé conjointement à `func_num_args` et `func_get_args` pour permettre aux fonctions utilisateurs d'accepter un nombre variable d'arguments.

8.41.12 `function_exists()` : Indique si une fonction est définie

bool `function_exists` (string `function_name`)

`function_exists` vérifie la liste des fonctions définies par l'utilisateur. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec `function_exists`

```
<?php
if (function_exists('imap_open')) {
    echo "Les fonctions IMAP sont disponibles.<br />\n";
} else {
    echo "Les fonctions IMAP ne sont pas disponibles.<br />\n";
}
?>
```

Notez qu'une fonction peut exister, même si elle est indisponible, à cause de la configuration ou des options de compilation. Notez également que `function_exists` retournera FALSE pour les instructions comme `include_once` et `echo`.

Voir aussi `method_exists`, `is_callable` et `get_defined_functions`.

8.41.13 `get_defined_functions()` : Liste toutes les fonctions définies

array `get_defined_functions` (void)

`get_defined_functions` retourne un tableau multi-dimensionnel, contenant la liste de toutes les fonctions définies, aussi bien les fonctions internes à PHP que celle déjà définie par l'utilisateur. Les noms des fonctions internes sont accessibles via `$arr["internal"]`, et les fonctions utilisateur sont accessibles via `$arr["user"]`.

Exemple avec `get_defined_functions`

```
<?php
function maligne($id, $data)
{
    return "<tr><th>$id</th><td>$data</td></tr>\n";
}

$arr = get_defined_functions();

print_r($arr);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [internal] => Array
```

```
(
[0] => zend_version
[1] => func_num_args
[2] => func_get_arg
[3] => func_get_args
[4] => strlen
[5] => strcmp
[6] => strncmp
...
[750] => bcscale
[751] => bccomp
)
[user] => Array
(
[0] => maligne
)
)
```

Voir aussi [function_exists](#) , [get_defined_vars](#) et [get_defined_constants](#) .

8.41.14 `register_shutdown_function()` : Enregistre une fonction pour exécution à l'extinction

`void register_shutdown_function (callback function , mixed parameter , mixed ...)`

`register_shutdown_function` enregistre la fonction `func` pour exécution à l'extinction du script.

Plusieurs appels à `register_shutdown_function` sont possibles dans le même script, et les fonctions seront appelées dans le même ordre que celui dans lequel elles sont enregistrées. Si vous appelez `exit` durant l'une des fonctions d'extinction, le processus sera définitivement arrêté, sans que les autres fonctions soient appelées.

En PHP 4.0.6 et précédent sous Apache, les fonctions d'extinction sont appelées après la fin de la requête (notamment les buffers ont été vidés), ce qui fait qu'il est impossible d'afficher du texte depuis cette fonction, avec `echo` ou `print` , ou encore de lire le contenu des buffers avec `ob_get_contents` . Depuis PHP 4.1, les fonctions d'extinction sont appelées comme une partie de la requête, donc, il est possible d'envoyer l'affichage depuis ces dernières. Il n'y a actuellement aucune façon de traiter les données avec les fonctions d'affichage du buffer dans la fonction d'extinction. La fonction d'extinction est appelé après avoir fermée tous les buffers de sortie ouverts, de ce fait, par exemple, la sortie ne sera pas compressée si `zlib.output_compression` est activé.

Depuis PHP 4, il est possible de passer des paramètres à la fonction d'extinction en passant des paramètres additionnels à la fonction `register_shutdown_function` .

Note

Typiquement, les fonctions non définies causent des erreurs fatales en PHP, mais lorsque la fonction `function` est appelée avec `register_shutdown_function` en étant non définie, une erreur de type `E_WARNING` est générée. De plus, pour des raisons internes à PHP cette erreur va être indiquée comme `Unknown at line #0` .

Note

Le dossier de travail du script peut changer dans la fonction d'extinction sous quelques serveurs web, e.g. Apache.

Note

La fonction d'extinction est appelé durant l'arrêt du script, donc, les en-têtes sont toujours déjà envoyés.

Voir aussi [auto_append_file](#) , [exit](#) , et la section sur la [gestion des connexions](#) .

8.41.15 `register_tick_function()` : Enregistre une fonction exécutée à chaque tick

`bool register_tick_function (callback function , mixed arg , mixed ...)`

`register_tick_function` enregistre la fonction `func` pour être exécutée à chaque fois qu'un `tick` est appelé.

Exemple avec `register_tick_function`

```
<?php
// Utilisation d'une fonction de callback
register_tick_function('ma_function', TRUE);

// Utilisation d'une méthode d'objet
$objet = new ma_classe();
register_tick_function(array(&$objet, 'ma_methode'), TRUE);
?>
```

Attention

`register_tick_function` ne doit pas être utilisé avec les modules de serveur web threadé. Les ticks ne fonctionnent pas en mode ZTS et peuvent interrompre votre serveur web.

Voir aussi [declare](#) et [unregister_tick_function](#) .

8.41.16 `unregister_tick_function()` : Annule la fonction exécutée à chaque tick

`void unregister_tick_function (string function_name)`

`unregister_tick_function` annule l'exécution automatique de `func` à chaque `tick` .

8.42 Gettext (GNU)

8.42.1 Introduction

Les fonctions gettext implémentent l'API NLS (Native Language Support) qui peut servir à internationaliser vos scripts PHP. Lisez la documentation GNU sur le site <http://www.gnu.org/software/gettext/manual/gettext.html> , ou bien sur votre système pour plus d'explications sur ces fonctions.

8.42.2 Pré-requis

Pour utiliser ces fonctions, vous devez télécharger et installer les bibliothèques GNU gettext depuis <http://www.gnu.org/software/gettext/gettext.html>

8.42.3 Installation

Pour inclure le module GNU gettext dans PHP, vous le compilez avec l'option `--with-gettext[=DIR]` , où DIR est le dossier d'installation de gettext. Par défaut, c'est `/usr/local` .

8.42.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.42.5 Types de ressources

Cette extension ne définit aucune ressource.

8.42.6 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [bind_textdomain_codeset](#)
- [bindtextdomain](#)
- [dcgettext](#)
- [dcngettext](#)
- [dgettext](#)
- [dngettext](#)
- [gettext](#)
- [ngettext](#)
- [textdomain](#)

8.42.8 bindtextdomain() : Fixe le chemin d'un domaine

string **bindtextdomain** (string domain , string directory)

bindtextdomain fixe le chemin du domaine domain à directory . Cette fonction retourne le chemin complet pour le domaine couramment utilisé.

Exemple avec bindtextdomain

```
<?php
$domain = 'monappli';
echo bindtextdomain($domain, '/usr/share/monappli/locale');
?>
```

L'exemple ci-dessus va afficher :

```
/usr/share/monappli/locale
```

8.42.9 dcgettext() : Remplace le domaine lors d'une recherche

string **dcgettext** (string domain , string message , int category)

dcgettext permet de remplacer le domaine courant lors de la recherche d'un message. Elle permet aussi de spécifier une catégorie category .

Voir aussi gettext .

8.42.10 dcngettext() : Version plurielle de dcgettext

string **dcngettext** (string domain , string msgid1 , string msgid2 , int n , int category)

dcngettext vous permet de remplacer le domaine courant domain pour une recherche simple au pluriel d'un message. Vous pouvez également spécifier une catégorie category .

Voir aussi ngettext .

8.42.11 dgettext() : Remplace le domaine courant

string **dgettext** (string domain , string message)

dgettext remplace le domaine courant domain pour une recherche simple dans message .

Voir aussi gettext .

8.42.12 dngettext() : Version plurielle de dgettext

string **dngettext** (string domain , string msgid1 , string msgid2 , int n)

dngettext vous permet de remplacer le domaine courant domain pour une recherche simple au pluriel d'un message.

Voir aussi ngettext .

8.42.13 gettext() : Recherche un message dans le domaine courant

string **gettext** (string message)

gettext retourne une chaîne traduite, si elle en a trouvé une dans la table de traduction, ou bien le message message , s'il n'a pas été trouvé. Vous pouvez utiliser le caractère souligné (_) comme alias de cette fonction.

Vérification gettext

```
<?php
// Choix l'allemand
setlocale(LC_ALL, 'de_DE');

// Spécifie la localisation des tables de traduction
bindtextdomain("myPHPApp", "./locale");

// Choisit le domaine
textdomain("myPHPApp");

// La traduction est cherché dans ./locale/de_DE/LC_MESSAGES/myPHPApp.mo

// Affichage d'un message de test
echo gettext("Bienvenue dans mon application PHP");

// Ou utilisation de l'alias _() pour gettext()
echo _("Passer une bonne journée");
?>
```

Voir aussi setlocale .

8.42.14 ngettext() : Version plurielle de gettext

string **ngettext** (string msgid1 , string msgid2 , int n)

ngettext retourne un message au pluriel identifié par msgid1 et msgid2 pour le compteur n . Quelques langues ont plus d'une forme de messages pluriels dépendant du compteur.

Exemple avec ngettext

```
<?php

setlocale(LC_ALL, 'cs_CZ');
printf(ngettext("%d window", "%d windows", 1), 1); // 1 okno
printf(ngettext("%d window", "%d windows", 2), 2); // 2 okna
printf(ngettext("%d window", "%d windows", 5), 5); // 5 oken

?>
```

8.42.15 `textdomain()` : Fixe le domaine par défaut

string **textdomain** (string text_domain)

textdomain fixe le domaine `text_domain` à utiliser lors des recherches avec gettext . Ce domaine dépend généralement de l'application.

8.43 Fonctions GMP

8.43.1 Introduction

Ces fonctions vous permettent de travailler avec des nombres de taille arbitraire, en utilisant la bibliothèque GNU MP .

Ces fonctions ont été ajoutées en PHP 4.0.4.

Note
La majorité des fonctions GMP acceptent des nombres GMP comme arguments, définis ci-dessous comme ressource . Cependant, la plupart de ces fonctions acceptent aussi des nombres et des chaînes à partir du moment où on peut les convertir en nombre. Si une fonction utilisant les entiers est plus rapide, elle sera automatiquement appelée si les arguments fournis sont des entiers. Cela se fait de manière transparente : vous pouvez donc utiliser des entiers avec les fonctions GMP sans perte de vitesse. Voir aussi gmp_init .
Attention
Si vous voulez explicitement spécifier un entier de grande taille, spécifiez-le sous forme de chaîne. Si vous ne le faites pas, PHP va interpréter votre entier et le transformer en une représentation interne, qui vous fera sûrement perdre de la précision, avant même que GMP n'entre en jeu.
Note
Cette extension est disponible sur les plates-formes Windows depuis PHP 5.1.0.

8.43.2 Pré-requis

Vous pouvez télécharger GMP sur le site de <http://www.swox.com/gmp/> . Ce site propose aussi un manuel GMP .

Vous devez utiliser GMP 2 ou plus récent pour utiliser ces fonctions. Certaines d'entre elles peuvent requérir une version encore plus récente de GMP .

8.43.3 Installation

Pour pouvoir utiliser ces fonctions, vous devez compiler PHP GMP en utilisant l'option `--with-gmp` .

8.43.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.43.5 Types de ressources

Cette extension ne définit aucune ressource.

8.43.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

GMP_ROUND_ZERO ([entier](#))
 GMP_ROUND_PLUSINF ([entier](#))
 GMP_ROUND_MINUSINF ([entier](#))

8.43.7 Exemples

Factorielle avec GMP
<pre><?php function fact(\$x) { \$return = 1; for (\$i=2; \$i < \$x; \$i++) { \$return = gmp_mul(\$return, \$i); } return \$return; } echo gmp_strval(fact(1000)) . "\n"; ?></pre>

Cet exemple va calculer factorielle de 1000 (un grand nombre) très vite.

8.43.8 Voir aussi

D'autres fonctions mathématiques sont disponibles avec les extensions [Nombres de grande taille](#) [BCMath](#) et [Mathématiques](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [Voir aussi](#)
- [gmp_abs](#)
- [gmp_add](#)
- [gmp_and](#)
- [gmp_clrbit](#)
- [gmp_cmp](#)
- [gmp_com](#)
- [gmp_div_q](#)
- [gmp_div_qr](#)
- [gmp_div_r](#)
- [gmp_div](#)
- [gmp_divexact](#)
- [gmp_fact](#)
- [gmp_gcd](#)
- [gmp_gcdext](#)

- [gmp_hamdist](#)
- [gmp_init](#)
- [gmp_intval](#)
- [gmp_invert](#)
- [gmp_jacobi](#)
- [gmp_legendre](#)
- [gmp_mod](#)
- [gmp_mul](#)
- [gmp_neg](#)
- [gmp_or](#)
- [gmp_perfect_square](#)
- [gmp_popcount](#)
- [gmp_pow](#)
- [gmp_powm](#)
- [gmp_prob_prime](#)
- [gmp_random](#)
- [gmp_scan0](#)
- [gmp_scan1](#)
- [gmp_setbit](#)
- [gmp_sign](#)
- [gmp_sqrt](#)
- [gmp_sqrtrem](#)
- [gmp_strval](#)
- [gmp_sub](#)
- [gmp_xor](#)

8.43.10 [gmp_add\(\)](#) : Addition de 2 nombres GMP

resource [gmp_add](#) (resource a , resource b)

[gmp_add](#) additionne les nombres GMP a et b . Le résultat est un nombre GMP .

Exemple avec [gmp_add](#)

```
<?php
$sum = gmp_add("123456789012345", "76543210987655");
echo gmp_strval($sum) . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
2000000000000000
```

8.43.11 [gmp_and\(\)](#) : ET logique

resource [gmp_and](#) (resource a , resource b)

[gmp_and](#) calcule le ET logique de a et b .

Exemple avec [gmp_and](#)

```
<?php
$and1 = gmp_and("0xfffffffff4", "0x4");
$and2 = gmp_and("0xfffffffff4", "0x8");
```

```

echo gmp_strval($and1) . "\n";
echo gmp_strval($and2) . "\n";
?>

```

L'exemple ci-dessus va afficher :

```

4
0

```

8.43.12 gmp_clrbit() : Annule un bit

void **gmp_clrbit** (resource a , int index)

gmp_clrbit met le bit index à 0 dans le nombre GMP a . L'index commence à zéro.

Note

Contrairement à la plupart des autres fonctions GMP, **gmp_clrbit** doit être appelée avec une ressource GMP existant déjà (en utilisant **gmp_init** par exemple). Elle ne sera pas automatiquement créée.

Exemple avec **gmp_clrbit**

```

<?php
$a = gmp_init("0xff");
gmp_clrbit($a, 0); // l'index commence à zéro
echo gmp_strval($a) . "\n";
?>

```

L'exemple ci-dessus va afficher :

```

254

```

Voir aussi **gmp_setbit** .

8.43.13 gmp_cmp() : Compare des nombres GMP

int **gmp_cmp** (resource a , resource b)

gmp_cmp retourne une valeur positive si $a > b$, zéro si $a = b$ et négative si $a < b$.

Exemple avec **gmp_cmp**

```

<?php
$cmp1 = gmp_cmp("1234", "1000"); // plus grand
$cmp2 = gmp_cmp("1000", "1234"); // plus petit
$cmp3 = gmp_cmp("1234", "1234"); // égal

echo "$cmp1 $cmp2 $cmp3\n";
?>

```

L'exemple ci-dessus va afficher :

```

1 -1 0

```

8.43.14 `gmp_com()` : Calcule le complémentaire d'un nombre

resource `gmp_com` (resource `a`)

Retourne le complémentaire de `a` .

Exemple avec `gmp_com`

```
<?php
$com = gmp_com("1234");
echo gmp_strval($com) . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
-1235
```

8.43.15 `gmp_div_q()` : Divisions de 2 nombres GMP

resource `gmp_div_q` (resource `a` , resource `b` , int `round`)

`gmp_div_q` divise le nombre GMP `b` par `a` . Le résultat est un entier. L'arrondi du résultat est défini par `round` , qui peut prendre l'une des valeurs suivantes :

- `gmp_round_zero` : Le résultat est tronqué vers 0.
- `gmp_round_plusinf` : Le résultat est tronqué vers +infinity
- `gmp_round_minusinf` : Le résultat est tronqué vers -infinity

`gmp_div_q` peut aussi être appelée `gmp_div` .

Exemple avec `gmp_div_q`

```
<?php
$div1 = gmp_div_q("100", "5");
echo gmp_strval($div1) . "\n";

$div2 = gmp_div_q("1", "3");
echo gmp_strval($div2) . "\n";

$div3 = gmp_div_q("1", "3", GMP_ROUND_PLUSINF);
echo gmp_strval($div3) . "\n";

$div4 = gmp_div_q("-1", "4", GMP_ROUND_PLUSINF);
echo gmp_strval($div4) . "\n";

$div5 = gmp_div_q("-1", "4", GMP_ROUND_MINUSINF);
echo gmp_strval($div5) . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
20
0
1
0
```

1

Voir aussi [gmp_div_r](#) et [gmp_div_qr](#) .

8.43.16 [gmp_div_qr\(\)](#) : Divise deux nombres GMP

array [gmp_div_qr](#) (resource *n* , resource *d* , int *round*)

[gmp_div_qr](#) divise *n* par *d* et retourne un tableau, dont le premier élément est $[n/d]$ (le quotient entier de la division) et le second est $(n - [n/d] * d)$ (le reste).

Voir [gmp_div_q](#) pour une description du paramètre *round* .

Division de nombres GMP

```
<?php
$a = gmp_init("0x41682179fbf5");
$res = gmp_div_qr($a, "0xDEF75");
printf("Le résultat est : q - %s, r - %s",
      gmp_strval($res[0]), gmp_strval($res[1]));
?>
```

Voir aussi [gmp_div_q](#) et [gmp_div_r](#) .

8.43.17 [gmp_div_r\(\)](#) : Reste de la division de deux nombres GMP

resource [gmp_div_r](#) (resource *n* , resource *d* , int *round*)

[gmp_div_r](#) le reste de la division entière de *n* par *d* . Le reste a le même signe que *n* , s'il est non nul.

Exemple avec [gmp_div_r](#)

```
<?php
$div = gmp_div_r("105", "20");
echo gmp_strval($div) . "\n";
?>
```

L'exemple ci-dessus va afficher :

5

Voir [gmp_div_q](#) pour une description du paramètre *round* .

Voir aussi [gmp_div_q](#) et [gmp_div_qr](#) .

8.43.18 [gmp_div\(\)](#) : Alias de [gmp_div_q](#)

Cette fonction est un alias de : [gmp_div_q](#) .

8.43.19 gmp_divexact() : Division exacte de nombres GMP

resource **gmp_divexact** (resource n , resource d)

gmp_divexact divise n par d , en utilisant les algorithmes de "division exacte". Cette fonction ne fournit de résultats cohérents que s'il est su par avance que d divise n .

Exemple avec **gmp_divexact**

```

<?php
$div1 = gmp_divexact("10", "2");
echo gmp_strval($div1) . "\n";

$div2 = gmp_divexact("10", "3"); // résultat incohérent
echo gmp_strval($div2) . "\n";
?>

```

L'exemple ci-dessus va afficher :

```

5
2863311534

```

8.43.20 gmp_fact() : Factorielle GMP

resource **gmp_fact** (int a)

gmp_fact calcule la factorielle de a : a! .

Exemple avec **gmp_fact**

```

<?php
$fact1 = gmp_fact(5); // 5 * 4 * 3 * 2 * 1
echo gmp_strval($fact1) . "\n";

$fact2 = gmp_fact(50); // 50 * 49 * 48, ... etc
echo gmp_strval($fact2) . "\n";
?>

```

L'exemple ci-dessus va afficher :

```

120
30414093201713378043612608166064768844377641568960512000000000000

```

8.43.21 gmp_gcd() : PGCD

resource **gmp_gcd** (resource a , resource b)

gmp_gcd calcule le PGCD (plus grand commun diviseur) de a et b . Le résultat est toujours positif, même si l'un des deux (ou les deux) nombres est négatif.

Exemple avec **gmp_gcd**

```

<?php
$gcd = gmp_gcd("12", "21");
echo gmp_strval($gcd) . "\n";

```

```
?>
```

L'exemple ci-dessus va afficher :

```
3
```

8.43.22 gmp_gcdext() : PGCD étendu

array **gmp_gcdext** (resource a , resource b)

gmp_gcdext calcule les entiers g, s, et t, tels que $a*s + b*t = g = \text{gcd}(a,b)$, où gcd est le pgcd de a et b. La fonction retourne un tableau avec les index g, s et t.

Cette fonction peut être utilisée pour résoudre des équations diophantines linéaires à deux variables. Ces équations n'ont qu'une seule solution entière, et elles sont de la forme : $a*x + b*y = c$. Pour plus d'informations, voyez les pages "[Diophantine Equation](#)" sur MathWorld, en anglais.

Résolution d'une équation Diophantine linéaire

```
// Résolution de l'équation a*s + b*t = g
// où a = 12, b = 21, g = gcd(12, 21) = 3
$a = gmp_init(12);
$b = gmp_init(21);
$g = gmp_gcd($a, $b);
$r = gmp_gcdext($a, $b);

$check_gcd = (gmp_strval($g) == gmp_strval($r['g']));
$eq_res = gmp_add(gmp_mul($a, $r['s']), gmp_mul($b, $r['t']));
$check_res = (gmp_strval($g) == gmp_strval($eq_res));

if ($check_gcd && $check_res) {
    $fmt = "Solution: %d*d + %d*d = %d\n";
    printf($fmt, gmp_strval($a), gmp_strval($r['s']), gmp_strval($b),
        gmp_strval($r['t']), gmp_strval($r['g']));
} else {
    echo "Erreur lors de la résolution de l'équation\n";
}

// Résultat : Solution : 12*2 + 21*-1 = 3
```

8.43.23 gmp_hamdist() : Distance de Hamming

int **gmp_hamdist** (resource a , resource b)

gmp_hamdist retourne la distance de Hamming entre a et b. Les deux paramètres doivent être strictement positifs.

Exemple avec gmp_hamdist

```
<?php
$ham1 = gmp_init("1001010011", 2);
$ham2 = gmp_init("1011111100", 2);
echo gmp_hamdist($ham1, $ham2) . "\n";

/* hamdist est équivalent à : */
echo gmp_popcount(gmp_xor($ham1, $ham2)) . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
6
6
```

Voir aussi [gmp_popcount](#) et [gmp_xor](#) .

8.43.24 [gmp_init\(\)](#) : Crée un nombre GMP

resource [gmp_init](#) (mixed number , int base)

[gmp_init](#) crée un nombre GMP, à partir d'un entier ou d'une chaîne. Les chaînes peuvent être en décimal ou en hexadécimal. Dans ce dernier cas, la chaîne doit commencer par 0x . Le paramètre optionnel base est disponible depuis PHP 4.1.0.

Création d'un nombre GMP

```
<?php
$a = gmp_init(123456);
$b = gmp_init("0xFFFFDEBACDFEDF7200");
?>
```

Note

Il n'est pas nécessaire d'appeler [gmp_init](#) si vous voulez utiliser des entiers ou des chaînes à la place de nombre GMP dans les fonctions GMP, comme [gmp_add](#) . Les arguments de ces fonctions sont automatiquement convertis en nombres GMP, si cette conversion est possible et nécessaire, en utilisant les mêmes règles que [gmp_init](#) .

8.43.25 [gmp_intval\(\)](#) : Convertit un nombre GMP en entier

int [gmp_intval](#) (resource gmpnumber)

[gmp_intval](#) convertit un nombre GMP en entier.

Attention

[gmp_intval](#) retourne un résultat cohérent uniquement si le nombre GMP peut être représenté par un entier PHP (i.e. type long signé). Si vous voulez simplement afficher un nombre GMP, utilisez plutôt [gmp_strval](#) .

Exemple avec [gmp_intval](#)

```
<?php
// affiche le bon résultat
echo gmp_intval("2147483647") . "\n";

// affiche un mauvais résultat, hors de portée des entiers PHP
echo gmp_intval("2147483648") . "\n";

// affiche le bon résultat
echo gmp_strval("2147483648") . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
2147483647
```

2147483647
2147483648

8.43.26 gmp_invert() : Inverse modulo

resource **gmp_invert** (resource a , resource b)

gmp_invert calcule l'inverse de a modulo b . **gmp_invert** retourne FALSE si un tel inverse n'existe pas.

Exemple avec **gmp_invert**

```
<?php
echo gmp_invert("5", "10"); // pas d'inverse, n'affiche rien, le résultat est FALSE
$invert = gmp_invert("5", "11");
echo gmp_strval($invert) . "\n";
?>
```

L'exemple ci-dessus va afficher :

9

8.43.27 gmp_jacobi() : Symbole de Jacobi

int **gmp_jacobi** (resource a , resource p)

gmp_jacobi calcule le symbole de Jacobi de a et p . p doit être positif et impair.

Exemple avec **gmp_jacobi**

```
<?php
echo gmp_jacobi("1", "3") . "\n";
echo gmp_jacobi("2", "3") . "\n";
?>
```

L'exemple ci-dessus va afficher :

1
0

8.43.28 gmp_legendre() : Symbole de Legendre

int **gmp_legendre** (resource a , resource p)

gmp_legendre calcule le symbole de Legendre de a et p . p doit être positif et impair.

Exemple avec **gmp_legendre**

```
<?php
echo gmp_legendre("1", "3") . "\n";
echo gmp_legendre("2", "3") . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
1
0
```

8.43.29 `gmp_mod()` : Modulo GMP

resource `gmp_mod` (resource `n` , resource `d`)

`gmp_mod` calcule `n` modulo `d` . Le résultat est toujours positif ou nul, car le signe de `d` est ignoré.

Exemple avec `gmp_mod`

```
<?php
$mod = gmp_mod("8", "3");
echo gmp_strval($mod) . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
2
```

8.43.30 `gmp_mul()` : Multiplication de 2 nombres GMP

resource `gmp_mul` (resource `a` , resource `b`)

`gmp_mul` multiplie les nombres GMP `a` et `b` .

Exemple avec `gmp_mul`

```
<?php
$mul = gmp_mul("12345678", "2000");
echo gmp_strval($mul) . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
24691356000
```

8.43.31 `gmp_neg()` : Opposé de nombre GMP

resource `gmp_neg` (resource `a`)

`gmp_neg` retourne l'opposé de `a` ($-1 * a$).

Exemple avec `gmp_neg`

```
<?php
$neg1 = gmp_neg("1");
echo gmp_strval($neg1) . "\n";
$neg2 = gmp_neg("-1");
echo gmp_strval($neg2) . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
-1
1
```

8.43.32 `gmp_or()` : OU logique

resource `gmp_or` (resource `a` , resource `b`)

`gmp_or` calcule le OU logique de `a` et `b` .

Exemple avec `gmp_or`

```
<?php
$or1 = gmp_or("0xffffffff2", "4");
echo gmp_strval($or1, 16) . "\n";
$or2 = gmp_or("0xffffffff2", "2");
echo gmp_strval($or2, 16) . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
ffffff6
ffffff2
```

8.43.33 `gmp_perfect_square()` : Carré parfait GMP

bool `gmp_perfect_square` (resource `a`)

`gmp_perfect_square` retourne TRUE si `a` est un carré parfait, et FALSE sinon.

Exemple avec `gmp_perfect_square`

```
<?php
// 3 * 3, carré parfait
var_dump(gmp_perfect_square("9"));

// pas un carré parfait
var_dump(gmp_perfect_square("7"));

// 1234567890 * 1234567890, carré parfait
var_dump(gmp_perfect_square("1524157875019052100"));
?>
```

L'exemple ci-dessus va afficher :

```
bool(true)
bool(false)
bool(true)
```

Voir aussi `gmp_sqrt` et `gmp_sqrtrem` .

8.43.34 gmp_popcount() : Compte de population

int **gmp_popcount** (resource a)

gmp_popcount dénombre la population de a .

Exemple avec **gmp_popcount**

```
<?php
$pop1 = gmp_init("10000101", 2); // 3 1's
echo gmp_popcount($pop1) . "\n";
$pop2 = gmp_init("11111110", 2); // 7 1's
echo gmp_popcount($pop2) . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
3
7
```

8.43.35 gmp_pow() : Puissance

resource **gmp_pow** (resource base , int exp)

gmp_pow élève base à la puissance exp . Dans le cas de 0^0 , **gmp_pow** retourne 1. exp ne doit pas être négatif.

Exemple avec **gmp_pow**

```
<?php
$pow1 = gmp_pow("2", 31);
echo gmp_strval($pow1) . "\n";
$pow2 = gmp_pow("0", 0);
echo gmp_strval($pow2) . "\n";
$pow3 = gmp_pow("2", -1); // Négatif, génère une alerte
echo gmp_strval($pow3) . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
2147483648
1
```

8.43.36 gmp_powm() : Puissance et modulo

resource **gmp_powm** (resource base , resource exp , resource mod)

gmp_powm calcule (base puissance exp) modulo mod . Si exp est négatif, le résultat est indéfini.

Exemple avec **gmp_powm**

```
<?php
$pow1 = gmp_powm("2", "31", "2147483649");
echo gmp_strval($pow1) . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
2147483648
```

8.43.37 `gmp_prob_prime()` : Nombre GMP probablement premier

`int gmp_prob_prime (resource a , int reps)`

Si `gmp_prob_prime` retourne 0, `a` est défini comme non premier. Si `gmp_prob_prime` retourne 1, alors `a` est "probablement" premier. Si `gmp_prob_prime` retourne 2, alors `a` est sûrement premier. `reps` peut raisonnablement varier de 5 à 10 (par défaut, c'est 10); une valeur supérieure réduit la probabilité qu'un nombre non premier soit identifié comme "probablement" premier.

`gmp_prob_prime` utilise le test de probabilité Miller-Rabin.

Exemple avec `gmp_prob_prime`

```
<?php
// pas premier
echo gmp_prob_prime("6") . "\n";

// probablement premier
echo gmp_prob_prime("111111111111111111") . "\n";

// premier
echo gmp_prob_prime("11") . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
0
1
2
```

8.43.38 `gmp_random()` : Nombre GMP aléatoire

`resource gmp_random (int limiter)`

`gmp_random` génère un nombre aléatoire. Ce nombre sera compris entre zéro et le nombre de bits par limb multiplié par `limiter`. Si `limiter` est négatif, un nombre négatif est généré.

Un limb est un mécanisme interne de GMP. Le nombre de bits dans un limb n'est pas statique, et peut varier entre les systèmes. En général, le nombre de bits par limb est 16 ou 32, mais ce n'est pas garanti.

Exemple avec `gmp_random`

```
<?php
$rand1 = gmp_random(1); // nombre aléatoire de 0 à 1 * bits per limb
$rand2 = gmp_random(2); // nombre aléatoire de 0 à 2 * bits per limb

echo gmp_strval($rand1) . "\n";
echo gmp_strval($rand2) . "\n";
```

```
?>
```

Cet exemple peut afficher :

```
1915834968
8642564075890328087
```

8.43.39 gmp_scan0() : Recherche 0

int **gmp_scan0** (resource a , int start)

gmp_scan0 recherche dans a , en commençant à la position start , vers les bits de poids lourd, jusqu'à ce qu'elle rencontre un bit à 0. Sa position est alors retournée.

Exemple avec **gmp_scan0**

```
<?php
// le bit "0" est trouvé à la position 3. l'index commence à 0
$s1 = gmp_init("10111", 2);
echo gmp_scan0($s1, 0) . "\n";

// le bit "0" est trouvé à la position 7. l'index commence à 5
$s2 = gmp_init("101110000", 2);
echo gmp_scan0($s2, 5) . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
3
7
```

8.43.40 gmp_scan1() : Recherche 1

int **gmp_scan1** (resource a , int start)

gmp_scan1 recherche dans a , en commençant à la position start , vers les bits de poids lourd, jusqu'à ce qu'elle rencontre un bit à 1. Sa position est alors retournée. Si aucun bit n'est trouvé, **gmp_scan1** retourne -1.

Exemple avec **gmp_scan1**

```
<?php
// "1" bit est trouvé à la position 3. l'index commence à 0
$s1 = gmp_init("01000", 2);
echo gmp_scan1($s1, 0) . "\n";

// "1" bit est trouvé à la position 9. l'index commence à 5
$s2 = gmp_init("01000001111", 2);
echo gmp_scan1($s2, 5) . "\n";
?>
```

L'exemple ci-dessus va afficher :

8.43.41 `gmp_setbit()` : Modifie un bit

void `gmp_setbit` (resource `a` , int `index` , bool `set_clear`)

`gmp_setbit` modifie le bit `index` dans `a` .

Note

Contrairement à la plupart des autres fonctions GMP, `gmp_setbit` doit être appelée avec une ressource GMP existant déjà (en utilisant `gmp_init` par exemple). Elle ne sera pas automatiquement créée.

Exemple avec `gmp_setbit`

```
<?php
$a = gmp_init("0xfd");
gmp_setbit($a, 1); // l'index commence à zéro
echo gmp_strval($a) . "\n";
?>
```

L'exemple ci-dessus va afficher :

255

Voir aussi `gmp_clrbit` .

8.43.42 `gmp_sign()` : Signe du nombre GMP

int `gmp_sign` (resource `a`)

`gmp_sign` retourne le signe de `a` : 1 si `a` est positif, -1 s'il est négatif et 0 si `a` est égal à zéro.

Exemple avec `gmp_sign`

```
<?php
// positive
echo gmp_sign("500") . "\n";

// negative
echo gmp_sign("-500") . "\n";

// zero
echo gmp_sign("0") . "\n";
?>
```

L'exemple ci-dessus va afficher :

1
-1
0

8.43.43 `gmp_sqrt()` : Racine carrée GMP

resource `gmp_sqrt` (resource *a*)

`gmp_sqrt` calcule la racine carrée de *a* .

Exemple avec `gmp_sqrt`

```
<?php
$sqrt1 = gmp_sqrt("9");
$sqrt2 = gmp_sqrt("7");
$sqrt3 = gmp_sqrt("1524157875019052100");

echo gmp_strval($sqrt1) . "\n";
echo gmp_strval($sqrt2) . "\n";
echo gmp_strval($sqrt3) . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
3
2
1234567890
```

8.43.44 `gmp_sqrtrem()` : Racine carrée avec reste GMP

array `gmp_sqrtrem` (resource *a*)

`gmp_sqrtrem` retourne un tableau dont le premier élément est la racine carrée entière de *a* (voir aussi `gmp_sqrt`), et le second est le reste de (i.e., la différence entre *a* et le carré du premier élément).

Exemple avec `gmp_sqrtrem`

```
<?php
list($sqrt1, $sqrt1rem) = gmp_sqrtrem("9");
list($sqrt2, $sqrt2rem) = gmp_sqrtrem("7");
list($sqrt3, $sqrt3rem) = gmp_sqrtrem("1048576");

echo gmp_strval($sqrt1) . ", " . gmp_strval($sqrt1rem) . "\n";
echo gmp_strval($sqrt2) . ", " . gmp_strval($sqrt2rem) . "\n";
echo gmp_strval($sqrt3) . ", " . gmp_strval($sqrt3rem) . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
3, 0
2, 3
1024, 0
```

8.43.45 `gmp_strval()` : Convertit un nombre GMP en chaîne

string `gmp_strval` (resource *gmpnumber* , int *base*)

`gmp_strval` convertit un nombre GMP en chaîne de caractères, dans la base `base` . La base par défaut est 10. Les valeurs possibles vont de 2 à 36.

Conversion d'un nombre GMP en chaîne

```
<?php
$a = gmp_init("0x41682179fbf5");
printf("Décimal : %s, base 36: %s",
      gmp_strval($a), gmp_strval($a, 36));
?>
```

8.43.46 `gmp_sub()` : Soustraction de 2 nombres GMP

resource `gmp_sub` (resource `a` , resource `b`)

`gmp_sub` soustrait le nombre GMP `b` de `a` . Le résultat est un nombre GMP.

Exemple avec `gmp_sub`

```
<?php
$sub = gmp_sub("281474976710656", "4294967296"); // 2^48 - 2^32
echo gmp_strval($sub) . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
281470681743360
```

8.43.47 `gmp_xor()` : OU exclusif logique

resource `gmp_xor` (resource `a` , resource `b`)

`gmp_or` calcule le OU exclusif logique de `a` et `b` .

Exemple avec `gmp_xor`

```
<?php
$xor1 = gmp_init("1101101110011101", 2);
$xor2 = gmp_init("0110011001011001", 2);

$xor3 = gmp_xor($xor1, $xor2);

echo gmp_strval($xor3, 2) . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
1011110111000100
```

8.44 HTTP

8.44.1 Introduction

Ces fonctions permettent de travailler sur les informations transmises au navigateur, via le protocole HTTP.

8.44.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.44.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.44.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.44.5 Types de ressources

Cette extension ne définit aucune ressource.

8.44.6 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [header](#)
- [headers_list](#)
- [headers_sent](#)
- [setcookie](#)
- [setrawcookie](#)

8.44.8 headers_list() : Retourne la liste des en-têtes de réponse du script courant

array headers_list (void)

`headers_list` retourne un tableau avec la liste des en-têtes qui seront transmis au navigateur. Pour déterminer si ces en-têtes ont été déjà envoyés ou pas, utilisez la fonction `headers_sent`.

Exemple avec `headers_list`

```
<?php

/* setcookie() va ajouter un en-tête */
setcookie('foo', 'bar');

/* Définit un en-tête de réponse
   Il sera ignoré par la majorité des navigateurs */
header("X-Sample-Test: foo");

/* Spécification de la réponse en texte simple */
header('Content-type: text/plain');

/* Quels sont les en-têtes qui vont être envoyés ? */
var_dump(headers_list());

?>
```

Affichera :

```
array(4) {
[0]=>
string(12) "X-Powered-By"
[1]=>
string(10) "Set-Cookie"
[2]=>
string(13) "X-Sample-Test"
[3]=>
string(12) "Content-type"
}
```

Voir aussi `headers_sent`, `header` et `setcookie`.

8.44.9 `headers_sent()` : Indique si les en-têtes HTTP ont déjà été envoyés

`bool headers_sent (string file , int line)`

`headers_sent` retourne TRUE si les en-têtes HTTP ont déjà été envoyés, et FALSE sinon. Si les paramètres optionnels `file` et `line` sont fournis, `headers_sent` va placer le nom du fichier source et le numéro de ligne qui ont débuté l'affichage, dans les variables `file` et `line`.

Vous ne pouvez plus envoyer d'en-têtes avec la fonction `header` une fois que le bloc d'en-tête a été fermé. En utilisant cette fonction, vous pouvez au moins éviter de voir s'afficher les erreurs HTTP liées. Une autre option consiste à utiliser le `contrôle de sortie`.

Note

Les paramètres optionnels `file` et `line` ont été ajoutés en PHP 4.3.0.

Exemple avec `headers_sent`

```
<?php
```

```
// Si aucun en-tête n'a été envoyé, envoyons-en un
if (!headers_sent()) {
    header ('Location: http://www.example.com/');
    exit;
}

// Voici un exemple d'utilisation des paramètres optionnels de fichier et de ligne
// disponibles depuis PHP 4.3.0. Notez que $filename et $linenum sont
// transmis pour utilisation ultérieure. Ne les assignez pas avant de les utiliser.
if (!headers_sent($filename, $linenum)) {
    header ('Location: http://www.example.com/');
    exit;
}

// Vous allez probablement déclencher une erreur ici
} else {

    echo "Les en-têtes ont déjà été envoyés, depuis le fichier $filename à la ligne $linenum\n" .
        "Il est donc impossible de vous rediriger automatiquement, aussi veuillez
        cliquez <a href=\"http://www.example.com\">ici</a>.\n";

    exit;
}

?>
```

Voir aussi [ob_start](#) , [trigger_error](#) et [header](#) pour plus de détails sur les tenants et aboutissants.

8.44.10 setcookie() : Envoie un cookie

bool **setcookie** (string name , string value , int expire , string path , string domain , bool secure)

setcookie définit un cookie qui sera envoyé avec le reste des en-têtes. Les cookies doivent passer avant tout autre en-tête (c'est une restriction des cookies, pas de PHP). Cela vous impose d'appeler cette fonction avant toute balise <html> ou <head> . Si quelque chose a été envoyé avant l'appel à cette fonction, **setcookie** échouera et retournera FALSE . Si **setcookie** réussit, elle retournera TRUE . Cela n'indique pas si le client accepte ou pas le cookie.

Note

Depuis PHP 4, vous pouvez utiliser la bufferisation de sortie pour pouvoir envoyer du contenu avant d'appeler cette fonction, avec la contrepartie que toute votre page sera envoyée en une fois. Vous pouvez faire cela en appelant [ob_start](#) et [ob_end_flush](#) dans votre script, ou en activant la directive `output_buffering` dans votre fichier de configuration `php.ini` .

Tous les arguments sauf name (nom) sont optionnels. Si seul le nom est présent, le cookie portant ce nom sera supprimé du navigateur de l'internaute. Vous pouvez aussi utiliser une chaîne vide comme valeur, pour ignorer un argument. Comme l'argument `expire` est un entier, il ne peut pas être ignoré avec une chaîne vide, vous devez utiliser le zéro pour cela (**0**). Le tableau suivant explique chaque paramètre de la fonction **setcookie** . Veuillez lire "[Netscape cookie specification](#)" pour le fonctionnement de chaque paramètre de **setcookie** ainsi que la [RFC 2965](#) pour des compléments d'informations sur les cookies HTTP.

Paramètre	Description	Exemples
name	Le nom du cookie.	'cookienom' est appelé via <code>\$_COOKIE['cookienom']</code>
value	La valeur du cookie. Cette valeur est stocké sur l'ordinateur du client ; ne	Le paramètre name est le 'cookienom', cette valeur est retrouvé en utilisant

	stocker pas d'informations importantes.	\$_COOKIE['cookiename']
expire	Le temps après lequel le cookie expire. C'est un timestamp Unix, donc, ce sera un nombre de secondes depuis l'époque Unix (1 Janvier 1970). En d'autres mots, vous devriez fixer cette valeur à l'aide de la fonction <u>time</u> et en y ajoutant le nombre de secondes après lequel on veut que le cookie expire. Vous pouvez utiliser aussi <u>mktime</u> .	time()+60*60*24*30 fera expirer le cookie dans 30 jours. Si vous ne spécifiez pas ce paramètre, le cookie expirera à la fin de la session (lorsque le navigateur sera fermé).
path	Le chemin sur le serveur sur lequel le cookie sera disponible.	Si la valeur est '/' , le cookie sera disponible sur l'ensemble du domaine domain . Si la valeur est '/foo/' , le cookie sera uniquement disponible dans le répertoire /foo/ ainsi que tous ces sous-répertoires comme /foo/bar/ du domaine domain . La valeur par défaut est le répertoire courant où le cookie a été défini.
domain	Le domaine où le cookie est disponible.	Pour rendre le cookie disponible sur tous les sous-domaines de example.com, vous devez mettre la valeur '.example.com' . Le point (.) n'est pas requis mais est nécessaire pour la compatibilité avec encore plus de navigateurs. Positionnez le à www.example.com rendra le cookie disponible uniquement sur le sous-domaine www . Reportez-vous aux <u>spécifications</u> pour plus de détails.
secure	Indique si le cookie doit uniquement être transmis à travers une connexion sécurisée HTTPS. Lorsqu'il est positionné à TRUE , le cookie ne sera positionné uniquement si la connexion sécurisée existe. La valeur par défaut est FALSE .	0 ou 1

Une fois que le cookie a été placé, il est accessible dans les variables globales \$_COOKIE ou bien \$HTTP_COOKIE_VARS arrays. Notez que les autoglobales telles que \$_COOKIE sont disponibles en PHP depuis la version 4.1.0 . \$HTTP_COOKIE_VARS existe depuis PHP 3. Les valeurs de cookies existent aussi dans la variable \$_REQUEST .

Note
Si la directive PHP <u>register_globals</u> est positionnée à on , la valeur du cookie est aussi disponible dans une variable. Dans l'exemple ci-dessous, <u>\$TestCookie</u> existe. Il est vivement recommandé d'utiliser <u>\$_COOKIE</u> .

Erreurs communes :

- Les cookies ne seront accessibles qu'au chargement de la prochaine page, ou au rechargement de la page courante. Pour tester si un cookie a été défini avec succès, vérifiez la présence du cookie au prochain chargement de la page avant que le cookie n'expire. Le délai d'expiration est défini en utilisant le paramètre expire . Une façon simple de vérifier le positionnement du cookie est d'utiliser print_r(\$_COOKIE); .

- Les cookies doivent être effacés avec les mêmes paramètres que ceux utilisés lors de leur création. Si l'argument `value` est une chaîne vide ou vaut `FALSE` et quelques autres arguments sont exactement les mêmes que lors du positionnement du cookie, alors le cookie sera effacé du client.
- Du fait que l'assignation d'une valeur vallant `FALSE` à un cookie tente de l'effacer, vous ne devriez pas utiliser de booléen . A la place, utilisez **0** pour `FALSE` et **1** pour `TRUE` .
- Les noms des cookies peuvent être des tableaux de noms et seront disponibles dans vos scripts PHP sous la forme de tableaux mais des cookies différents seront placés sur le client. Utilisez explode pour placer un cookie avec des noms et des valeurs multiples. Il n'est pas recommandé d'utiliser la fonction serialize pour réaliser ceci, car cela peut conduire à des problèmes de sécurité.

En PHP 3, les appels multiples à setcookie dans le même script seront effectués dans l'ordre inverse. Si vous essayez d'effacer un cookie avant d'insérer une nouvelle valeur, vous devez placer l'insertion avant l'effacement. Depuis PHP 4, les appels multiples à setcookie sont effectués dans un ordre naturel.

Quelques exemples :

Exemples avec setcookie

```
<?php
$value = 'Valeur de test';

setcookie("TestCookie", $value);
setcookie("TestCookie", $value, time()+3600); /* expire dans une heure */
setcookie("TestCookie", $value, time()+3600, "/~rasmus/", ".utoronto.ca", 1);
?>
```

Notez que la partie "valeur" du cookie sera automatiquement encodée URL lorsque vous envoyez le cookie et, lorsque vous le recevez, il sera automatiquement décodé, et affecté à la variable du même nom que le cookie. Si vous le voulez pas de ce comportement par défaut, vous pouvez utiliser la fonction setrawcookie si vous utilisez PHP 5. Pour voir le résultat, essayez les scripts suivants :

Affectation des valeurs de cookie

```
<?php
// Afficher un cookie
echo $_COOKIE["TestCookie"];
echo $HTTP_COOKIE_VARS["TestCookie"];

// Une autre méthode pour afficher tous les cookies
print_r($_COOKIE);
?>
```

Lorsque vous effacez un cookie, vous devriez toujours vous assurer que sa date d'expiration est déjà passée, pour déclencher le mécanisme de votre navigateur. Voici comment procéder :

Exemple d'effacement de cookies avec setcookie

```
<?php
// utilisation de la date moins une heure
setcookie ("TestCookie", "", time() - 3600);
setcookie ("TestCookie", "", time() - 3600, "/~rasmus/", ".example.com", 1);
?>
```

Vous pouvez aussi utiliser les cookies avec des tableaux, en utilisant la notation des tableaux. Cela a pour effet de créer autant de cookies que votre tableau a d'éléments, mais lorsque les cookies seront reçus par votre script, les valeurs seront placées dans un tableau :

Utilisation des tableaux avec [setcookie](#)

```

<?php
setcookie("cookie[three]", "cookieethree" );
setcookie("cookie[two]", "cookietwo" );
setcookie("cookie[one]", "cookieone" );

// Après avoir rechargé la page :
if (isset($_COOKIE['cookie'])) {
    foreach ($_COOKIE['cookie'] as $name => $value) {
        echo "$name : $value <br />\n";
    }
}
?>

```

L'exemple ci-dessus va afficher :

```

three : cookieethree
two : cookietwo
one : cookieone

```

Note

Les RFCs suivantes peuvent être utiles : [RFC 2109](#) et [RFC 2695](#) .

Vous pourrez noter que le paramètre `expire` prend un timestamp unique, et non pas la date au format Jour, JJ-Mois-AAAA HH:MM:SS GMT , car PHP fait la conversion en interne.

Le paramètre `expire` est comparé avec le temps du client qui peut être différent de celui du serveur.

Note

Microsoft Internet Explorer 4 utilisé avec le Service Pack 1 ne gère pas bien les cookies qui possèdent un paramètre `path` .

Netscape Communicator 4.05 et Microsoft Internet Explorer 3.x semblent ne pas gérer correctement les cookies lorsque `path` et `expire` ne sont pas fournis.

Voir aussi [header](#) , [setrawcookie](#) ainsi que la [section sur les cookies](#) .

8.44.11 `setrawcookie()` : Envoie un cookie sans encoder sa valeur en URL

`bool setrawcookie (string name , string value , int expire , string path , string domain , bool secure)`

`setrawcookie` est exactement la même que `setcookie` excepté que la valeur du cookie ne sera pas automatiquement encodée URL lors de l'envoi au navigateur.

Voir aussi [header](#) , [setcookie](#) et la [section sur les cookies](#) .

8.45 Hyperwave

8.45.1 Introduction

Hyperwave a été développé par IICM à Graz. Son nom original était HYPER-G et il a pris le nom de Hyperwave lors de sa commercialisation (en 1996).

Hyperwave n'est pas gratuit. La version actuelle est la 5.5, disponible à <http://www.hyperwave.com/>. Une version limitée à 30 jours peut être demandée.

Voir aussi le module Hyperwave API.

HIS est un système d'information similaire à une base de données, (HIS, Hyperwave Information Server). HIS se concentre sur l'enregistrement et la gestion des documents. Un document peut être n'importe quelle donnée, qui peut être stockée dans un fichier. Chaque document est accompagné par un enregistrement. Cet enregistrement contient des méta données à propos du document. Ces méta données sont des listes d'attributs qui peuvent être étendues par l'utilisateur. Un attribut est une paire clé/valeur de la forme : clé =valeur. L'enregistrement complet contient autant de paires que le désire l'utilisateur. Le nom d'un attribut n'a pas besoin d'être unique, c'est-à-dire qu'une même clé peut apparaître plusieurs fois dans un enregistrement. Cela peut être utile si vous devez donner un titre à votre document en plusieurs langues, par exemple. Dans un cas pareil, la convention est que chaque valeur de titre est précédée par deux lettres et deux points, tel que : 'fr:Titre en français' ou 'ge:Titel in deutsch'. D'autres attributs comme une description ou des mots-clés sont aussi susceptibles de recourir à ce genre de procédé. Vous pouvez aussi remplacer l'abréviation de langage par une autre chaîne, tant qu'elle est séparée de la valeur par les deux points.

Chaque enregistrement a une représentation native qui contient toutes les paires clé/valeur, séparées par un retour à la ligne. L'extension Hyperwave reconnaît une autre représentation qui est un tableau associatif, où les attributs servent de clés. Les attributs multilingues étant gérés sous la forme d'un autre tableau associatif, dont les clés sont les chaînes de langue. En fait, tous les attributs multiformes sont gérés sous la forme de tableau associatif. (Cela n'est pas encore complètement codé. Uniquement les attributs de titre, description et mot-clés sont traités correctement).

En dehors des documents, tous les hyperliens contenus dans un document sont enregistrés dans un autre enregistrement. Les hyperliens qui sont à l'intérieur d'un document en seront supprimés, et enregistrés dans des objets particuliers au moment de l'insertion dans la base de données. L'enregistrement des hyperliens contient les informations d'origine et d'objectif. Afin d'accéder au document original, vous devrez lire le document sans les liens, puis lire les liens et les réinsérer (les fonctions hw_pipedocument et hw_gettext le font pour vous. L'avantage de séparer les liens du document est évident : une fois qu'un document, cible d'un hyperlien, a été renommé, le liens peut facilement être modifié. Le document contenant le lien n'est pas modifié pour autant. Vous pouvez même ajouter un lien à un document sans le modifier.

Dire que hw_pipedocument et hw_gettext font l'insertion automatiquement n'est pas aussi simple qu'il y paraît. L'insertion implique une certaine hiérarchie de documents. Sur un serveur web, la hiérarchie est fournie par le système de fichiers, mais Hyperwave dispose de sa propre hiérarchie et les noms de fichiers ne reflètent pas la position d'un objet dans cette hiérarchie. Ainsi, la création de liens requiert en premier lieu la construction de la hiérarchie et de l'espace des noms dans une hiérarchie web et un espace de nom web. La différence fondamentale entre Hyperwave et le web est qu'il y a une distinction claire entre les noms et la hiérarchie dans Hyperwave. Le nom ne contient aucune information à propos de la position de l'objet dans la hiérarchie. Sur le web, le nom

contient les informations de localisation dans la hiérarchie. Cela conduit à deux méthodes d'accès : soit la hiérarchie Hyperwave et le nom de l'objet sont inscrits dans l'URL. Pour simplifier les choses, une deuxième approche est pratiquée. L'objet Hyperwave de nom 'mon_objet' correspond à l'URL 'http://hote/mon_objet', peu importe alors où il est rangé dans la hiérarchie. Un objet dont le nom est 'parent/mon_objet' peut être le fils de l'objet 'mon_objet' dans la hiérarchie Hyperwave, bien que ce soit le contraire en convention web, et cela risque de perturber l'utilisateur.

Ayant pris cette décision, un deuxième problème surgit : comment faire l'interface avec PHP ? L'URL `http://hote/mon_objet` n'appellera aucun script PHP à moins que vous ne demandiez à votre serveur web de le remplacer par autre chose, comme : '`http://host/php3_script/mon_objet`' et le script 'php3_script' utilise la variable `$PATH_INFO` pour rechercher l'objet 'mon_objet' sur le serveur Hyperwave. Il y a juste un petit inconvénient, qui peut facilement être corrigé. Réécrire une URL ne vous permettra aucun accès aux autres documents du serveur web. Un script de recherche dans le serveur Hyperwave serait impossible. Il vous faudra donc au moins une autre règle pour exclure certaines URL, comme celles qui commencent par `http://host/Hyperwave`. Voici, de manière simple, comment partager un espace de nom entre un serveur web et un serveur Hyperwave serveur.

Basé sur le mécanisme précédent, on trouve l'insertion dans les documents.

Il est plus compliqué d'avoir PHP ne fonctionnant pas comme un module de serveur, ou un script CGI, mais comme une application indépendante. Dans ce cas, il est utile d'inscrire la hiérarchie et le nom de fichier Hyperwave dans le système de fichier. Mais comme cela risque d'entrer en conflit avec le séparateur de dossier ('/'), il faut le remplacer par un autre caractère, '.'.

Le protocole réseau pour communiquer avec un serveur Hyperwave est appelé HG-CSP (Hyper-G Client/Server Protocol). Il est basé sur des messages qui initient des actions, comme lire l'en-tête de fichier. Dans les premières versions de Hyperwave Server, deux clients natifs (Harmony, Amadeus) étaient fournis pour permettre la communication avec le serveur. Ils ont disparu lors de la commercialisation de Hyperwave. En tant qu'ersatz, un client appelé wavemaster est désormais fourni. wavemaster est un espèce de convertisseur de protocole de *HTTP* en *HG-CSP*. L'idée est de faire toute l'administration de la base et la visualisation des documents par une interface web. Le wavemaster implémente un jeu d'emplacement pour certaines actions de personnalisation de l'interface. Ce jeu est appelé *PLACE* langage. *PLACE* pêche encore par le manque de nombreuses fonctions de programmation et le manque d'évolutivité. Cela a conduit à l'utilisation de JavaScript ce qui ne rend pas la vie facile.

Que PHP supporte Hyperwave permet de combler ces manques. PHP implémente tous les messages définis par *HG-CSP* mais fourni d'autres commandes puissantes, comme celle de lire des documents complets.

Hyperwave dispose de sa propre terminologie pour localiser certaines informations. Cette terminologie a été largement étendue. Presque toutes les fonctions utilisent l'un des types de données suivants :

- object ID : un entier unique pour chaque objet sur le serveur Hyperwave. C'est aussi un des attributs de l'enregistrement de l'objet (ObjectID). Les object ids sont souvent utilisés comme paramètre d'entrée pour spécifier un objet.
- object record : une chaîne contenant des paires clé=valeur. Les paires sont séparées par un retour à la ligne. Un enregistrement d'objet peut facilement être converti en tableau, avec la fonction `hw_objrec2array`. De nombreuses fonctions retournent un objet record. Ces fonctions ont leur nom qui finit par `obj`.
- object array : un tableau associatif qui contient tous les attributs d'un objet. La clé est l'attribut. Si un attribut apparaît plusieurs fois, il sera représenté sous la forme d'un tableau associatif ou indexé. Les attributs qui dépendent des langues (comme `title`, `keyword` ou

description) seront représentés sous la forme d'un tableau associatif dont les clés seront les abréviations de langues. Tous les autres attributs à valeur multiple prendront la forme d'un tableau indexé.

- `hw_document` : ce type est un nouveau type de données, qui contient le document lui-même, comme HTML, PDF etc. Il est optimisé pour les documents HTML mais peut s'utiliser avec n'importe quel format.

De nombreuses fonctions qui retournent un tableau d'enregistrements, retournent aussi un tableau associé, avec des informations statistiques. Ce tableau est le dernier élément du tableau d'enregistrements. Les statistiques contiennent les entrées suivantes :

Hidden

Nombre d'objets dont l'attribut `PresentationHints` est Hidden.

CollectionHead

Nombre d'objets dont l'attribut `PresentationHints` est CollectionHead.

FullCollectionHead

Nombre d'objets dont l'attribut `PresentationHints` est FullCollectionHead.

CollectionHeadNr

Index du premier objet du tableau d'enregistrement avec l'attribut `PresentationHints` à CollectionHead.

FullCollectionHeadNr

Index du premier objet du tableau d'enregistrement avec l'attribut `PresentationHints` est FullCollectionHead.

Total

Total : nombre d'enregistrements.

8.45.2 Pré-requis

Cette extension requiert un serveur Hyperwave, téléchargeable sur le site de <http://www.hyperwave.com/>.

8.45.3 Installation

Cette extension [PECL](#) n'est pas intégrée à PHP.

Si vous voulez utiliser ces fonctions, vous devez compiler PHP avec le support Hyperwave en utilisant l'option de configuration `--with-hyperwave[=DIR]`.

Les utilisateurs de Windows doivent activer la bibliothèque `php_hyperwave.dll` dans le `php.ini` pour pouvoir utiliser ces fonctions. Vous pouvez télécharger ces bibliothèques DLL pour ces extensions PECL depuis la page [de téléchargement de PHP](#) sous sur la page <http://snaps.php.net/>.

8.45.4 Intégration avec Apache

L'extension Hyperwave est utilisée de manière optimale lorsque PHP est compilé comme module Apache. Dans ce cas, le serveur Hyperwave sous-jacent peut être caché quasiment totalement aux utilisateurs, si Apache utilise son moteur d'écriture. Les explications suivantes vous éclaireront :

Etant donné que PHP avec l'extension Hyperwave et Apache tend à remplacer la solution native basée sur Wavemaster, je vais supposer que le serveur Apache servira seulement d'interface Hyperwave. Ce n'est pas nécessaire, mais cela simplifie grandement la configuration. Le concept

est très simple. Premièrement, vous avez besoin d'un script PHP qui évalue la variable `$_ENV['PATH_INFO']` et considère que cette valeur est un objet Hyperwave. Appelons ce script 'Hyperwave'. L'URL `http://votre.hote/Hyperwave/nom_objet` retourne l'objet Hyperwave dont le nom est 'nom_objet'. Le script doit alors réagir suivant le type de l'objet. Si c'est un groupe, il devra probablement retourner une liste de fils. Si c'est un document, il pourra retourner son type MIME et son contenu. Une amélioration peut être obtenue en utilisant le moteur de réécriture d'Apache. D'un point de vue utilisateur, il est plus direct si l'URL `http://votre.hote/nom_objet` retourne l'objet. La règle de réécriture est simple :

Configuration d'Apache pour HyperWave
<code>RewriteRule ^/(.*) /usr/local/apache/htdocs/HyperWave/\$1 [L]</code>

Maintenant, toutes les URL pointent sur un objet Hyperwave. Cela conduit à un problème simple. Il n'y a pas d'autre façon d'exécuter, c'est-à-dire rechercher, un autre script que ce script 'Hyperwave'. Cela pourra être corrigé avec une autre règle telle que :

Configuration d'Apache pour HyperWave (2)
<code>RewriteRule ^/hw/(.*) /usr/local/apache/htdocs/hw/\$1 [L]</code>

Le dossier `/usr/local/apache/htdocs/hw` sera ainsi réservé pour d'autres scripts et fichiers. Assurez-vous que cette règle est évaluée avant la première règle que nous avons défini. Il y a juste un léger inconvénient : tous les objets Hyperwave qui commencent par 'hw/' seront cachés. Alors, assurez-vous que vous n'utilisez pas de tels noms. Si vous avez besoin d'autres dossiers, par exemple, un dossier d'images, ajoutez simplement d'autres règles. N'oubliez pas de lancer le moteur de réécriture avec.

Demarrer les règles de réécriture d'Apache pour HyperWave
<code>RewriteEngine on</code>

Mon expérience m'a montré que vous aurez besoin des scripts suivants :

- Retourne l'objet lui-même
- Pour autoriser la recherche
- S'identifier
- Choisir une configuration
- Un script pour chaque fonction supplémentaire, comme afficher un objet, afficher des informations sur les utilisateurs, afficher le statut du serveur, etc.

Une alternative pour le moteur de réécriture est l'utilisation de la directive Apache `ErrorDocument`, mais sachez alors que `ErrorDocument` n'effectue pas les redirections de méthodes POST.

8.45.5 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable	Historique
<code>hyperwave.allow_persistent</code>	"0"	PHP_INI_SYSTEM	Disponible depuis PHP 4.3.2.
<code>hyperwave.default_port</code>	"418"	PHP_INI_ALL	

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#).

8.45.6 Types de ressources

Cette extension ne définit aucune ressource.

8.45.7 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

HW_ATTR_LANG ([entier](#))

HW_ATTR_NR ([entier](#))

HW_ATTR_NONE ([entier](#))

8.45.8 A faire

Il reste encore beaucoup à faire :

- La fonction `hw_InsertDocument` doit être séparée en deux : `hw_insertobject()` et `hw_putdocument()`.
- Les noms de nombreuses fonctions ne sont pas encore fixés.
- La majorité des fonctions requièrent la connexion courante comme premier paramètre. Cela conduit à beaucoup de frappe clavier, même s'il n'y a souvent qu'une seule connexion en jeu. Une connexion par défaut améliorerait ceci.
- La conversion depuis un objet record en un objet array a besoin de gérer les attributs multiples.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Intégration avec Apache](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [A faire](#)
- [hw_Array2Objrec](#)
- [hw_changeobject](#)
- [hw_Children](#)
- [hw_ChildrenObj](#)
- [hw_Close](#)
- [hw_Connect](#)
- [hw_connection_info](#)
- [hw_cp](#)
- [hw_Deleteobject](#)
- [hw_DocByAnchor](#)
- [hw_DocByAnchorObj](#)
- [hw_Document_Attributes](#)
- [hw_Document_BodyTag](#)
- [hw_Document_Content](#)
- [hw_Document_SetContent](#)
- [hw_Document_Size](#)
- [hw_dummy](#)
- [hw_EditText](#)
- [hw_Error](#)
- [hw_ErrorMsg](#)
- [hw_Free_Document](#)
- [hw_GetAnchors](#)
- [hw_GetAnchorsObj](#)

- [hw_GetAndLock](#)
- [hw_GetChildColl](#)
- [hw_GetChildCollObj](#)
- [hw_GetChildDocColl](#)
- [hw_GetChildDocCollObj](#)
- [hw_GetObject](#)
- [hw_GetObjectByQuery](#)
- [hw_GetObjectByQueryColl](#)
- [hw_GetObjectByQueryCollObj](#)
- [hw_GetObjectByQueryObj](#)
- [hw_GetParents](#)
- [hw_GetParentsObj](#)
- [hw_getrellink](#)
- [hw_GetRemote](#)
- [hw_getremotechildren](#)
- [hw_GetSrcByDestObj](#)
- [hw_GetText](#)
- [hw_getusername](#)
- [hw_Identify](#)
- [hw_InCollections](#)
- [hw_Info](#)
- [hw_InsColl](#)
- [hw_InsDoc](#)
- [hw_insertanchors](#)
- [hw_InsertDocument](#)
- [hw_InsertObject](#)
- [hw_mapid](#)
- [hw_Modifyobject](#)
- [hw_mv](#)
- [hw_New_Document](#)
- [hw_objrec2array](#)
- [hw_Output_Document](#)
- [hw_pConnect](#)
- [hw_PipeDocument](#)
- [hw_Root](#)
- [hw_setlinkroot](#)
- [hw_stat](#)
- [hw_Unlock](#)
- [hw_Who](#)

8.45.10 [hw_changeobject\(\)](#) : Change les attributs d'un objet (obsolète)

bool [hw_changeobject](#) (int link , int objid , array attributes)

Attention
 Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.45.11 [hw_Children\(\)](#) : Liste des objets ids des objets fils

array [hw_children](#) (int connection , int objectID)

[hw_Children](#) retourne un tableau avec des objets ids. Chaque objet id est celui d'un des fils du groupe dont l'id est objectID . Ce tableau contient tous les fils, documents et groupes.

8.45.12 **hw_ChildrenObj()** : Liste des objets records des objets fils

array **hw_childrenobj** (int connection , int objectID)

hw_ChildrenObj retourne un tableau avec des objets records. Chaque objet records est celui d'un des fils du groupe dont l'id est objectID . Ce tableau contient tous les fils, documents et groupes

8.45.13 **hw_Close()** : Ferme la connexion Hyperwave

bool **hw_close** (int connection)

hw_Close ferme la connexion connection à un serveur Hyperwave. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.45.14 **hw_Connect()** : Ouvre une connexion Hyperwave

int **hw_connect** (string host , int port , *string* username , string password)

hw_Connect ouvre une connexion Hyperwave et retourne un identifiant de connexion en cas de succès, ou FALSE si la connexion n'a pas pu être créée. Chaque argument doit être entouré de guillemets, sauf le numéro de port. Les arguments username et password sont optionnels et peuvent être ignorés. Dans ce cas, aucune identification ne sera faite au niveau du serveur. Cela revient à s'identifier en tant qu'utilisateur anonyme. Cette fonction retourne un identifiant de connexion qui sera nécessaire aux autres fonctions Hyperwave. Vous pouvez avoir plusieurs connexions simultanées. N'oubliez pas que les mots de passe ne sont pas cryptés.

Voir aussi hw_pconnect .

8.45.15 **hw_connection_info()** : Affiche des informations sur la connexion au serveur Hyperwave

void **hw_connection_info** (int link)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.45.16 **hw_cp()** : Copie des objets HyperWave

int **hw_cp** (int connection , array object_id_array , int destination_id)

hw_cp copie les objets ayant les identifiants object_id_array , et crée un groupe ayant l'objet id destination_id .

La valeur retournée est le nombre d'objets copiés.

Voir aussi hw_mv .

8.45.17 **hw_Deleteobject() : Efface des objets**

bool **hw_deleteobject** (int connection , int object_to_delete)

hw_Deleteobject efface l'objet dont l'identifiant est object_to_delete . Toutes les instances de l'objet object_to_delete seront effacées.

hw_deleteobject retourne TRUE si aucune erreur ne survient, sinon, FALSE .

Voir aussi hw_mv .

8.45.18 **hw_DocByAnchor() : Identifiant d'objet de l'objet dans l'ancrage**

int **hw_docbyanchor** (int connection , int anchorID)

hw_DocByAnchor retourne l'identifiant d'objet de l'objet dans l'ancrage anchorID .

8.45.19 **hw_DocByAnchorObj() : Attributs de l'objet dans l'ancrage**

string **hw_docbyanchorobj** (int connection , int anchorID)

hw_DocByAnchorObj retourne les attributs du document qui correspond à anchorID .

8.45.20 **hw_Document_Attributes() : Object record de hw_document**

string **hw_document_attributes** (int hw_document)

hw_Document_Attributes retourne les attributs du document hw_document .

Voir aussi hw_document_bodytag et hw_document_size .

Pour la compatibilité ascendante, hw_documentattributes est aussi accepté, mais il est déconseillé.

Voir aussi hw_document_bodytag et hw_document_size .

8.45.21 **hw_Document_BodyTag() : Balise de corps d'un document**

string **hw_document_bodytag** (int hw_document , string prefix)

hw_document_bodytag retourne la balise BODY du document hw_document . Si le document est un document HTML, la balise BODY doit être affichée avant le document.

Voir aussi hw_document_attributes et hw_document_size .

Pour la compatibilité ascendante, hw_documentbodytag est aussi accepté, mais il est déconseillé.

8.45.22 `hw_Document_Content()` : Contenu d'un document

string `hw_document_content` (int `hw_document`)

`hw_documentcontent` retourne la balise BODY du document `hw_document` .Si le document est un document HTML, la balise BODY doit être affichée avant le document.

Voir aussi [hw_document_attributes](#) , [hw_document_size](#) et [hw_document_setcontent](#) .

8.45.23 `hw_Document_SetContent()` : Modifie/remplace le contenu d'un document

bool `hw_document_setcontent` (int `hw_document` , string `content`)

`hw_document_setcontent` modifie/remplace le contenu du document `hw_document` par la valeur de `content` . Si le document est un document HTML, le contenu représente tout ce qui est placé au-delà de la balise BODY. Les informations de HEAD et de la balise BODY sont enregistrées dans les attributs. Si vous fournissez aussi ces informations dans le corps du document, le serveur Hyperwave modifiera les attributs. Cela n'est cependant pas une bonne idée. Si la fonction échoue, l'ancien contenu est restauré.

Voir aussi [hw_document_attributes](#) , [hw_document_size](#) et [hw_document_content](#) .

8.45.24 `hw_Document_Size()` : Taille d'un document

int `hw_document_size` (int `hw_document`)

`hw_document_size` retourne la taille du document en octets.

Voir aussi [hw_document_bodytag](#) et [hw_document_attributes](#) .

Pour la compatibilité ascendante, [hw_documentsize](#) est aussi accepté, mais il est déconseillé.

8.45.25 `hw_dummy()` : Fonction sans objet pour Hyperwave

string `hw_dummy` (int `link` , int `id` , int `msgid`)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.45.26 `hw_EditText()` : Retourne un document texte

bool `hw_edittest` (int `connection` , int `hw_document`)

`hw_EditText` charge le texte du document sur le serveur. Les attributs du document ne doivent pas être modifiés tant que le document est en train d'être édité. Cette fonction n'est disponible que sur les documents texte. Elle n'ouvrira pas de canal de transfert, et donc, bloquera le script durant le transfert.

Voir aussi [hw_pipedocument](#) , [hw_free_document](#) , [hw_document_bodytag](#) , [hw_document_size](#) , [hw_output_document](#) et [hw_gettext](#) .

8.45.27 [hw_Error\(\)](#) : Retourne le code d'erreur

int [hw_error](#) (int connection)

[hw_Error](#) retourne le code d'erreur de la dernière erreur. Si la valeur 0 est retournée, c'est qu'il n'y avait pas d'erreur. L'erreur se rapporte à la dernière commande.

8.45.28 [hw_ErrorMsg\(\)](#) : Retourne un message d'erreur

string [hw_errormsg](#) (int connection)

[hw_ErrorMsg](#) retourne une chaîne contenant le dernier message d'erreur ou 'No Error' (pas d'erreur). Si FALSE est retourné, cette fonction a échoué. Ce message est relatif à la dernière commande exécutée.

8.45.29 [hw_Free_Document\(\)](#) : Détruit un document

bool [hw_free_document](#) (int hw_document)

[hw_Free_Document](#) détruit un document Hyperwave.

8.45.30 [hw_GetAnchors\(\)](#) : Identifiants des ancrages d'un document

array [hw_getanchors](#) (int connection , int objectID)

[hw_GetAnchors](#) retourne un tableau contenant les identifiants des ancrages du document objectID .

8.45.31 [hw_GetAnchorsObj\(\)](#) : Attributs des ancrages d'un document

array [hw_getanchorsobj](#) (int connection , int objectID)

[hw_getanchorsobj](#) retourne un tableau d'attributs des ancrages du document objectid .

8.45.32 [hw_GetAndLock\(\)](#) : Retourne les attributs et verrouille l'objet

string [hw_getandlock](#) (int connection , int objectID)

[hw_getandlock](#) retourne les attributs et verrouille l'objet objectID . Le verrouillage empêchera les autres utilisateurs d'y accéder jusqu'à ce qu'il soit deverrouillé.

Voir aussi [hw_unlock](#) et [hw_getobject](#) .

8.45.33 **hw_GetChildColl()** : Identifiant d'objets des groupes fils

array **hw_getchildcoll** (int connection , int objectID)

hw_getchildcoll retourne un tableau contenant les identifiants d'objets des groupes fils du groupe objectID . Cette fonction ne retournera pas d'identifiants d'objets des documents fils.

Voir aussi hw_children et hw_getchilddoccoll .

8.45.34 **hw_GetChildCollObj()** : Objet record d'un groupe d'enfants

array **hw_getchildcollobj** (int connection , int objectID)

hw_getchildcollobj retourne un tableau d'object record. Chaque object record appartient à un groupe d'enfants de la collection objectID . La fonction ne retournera pas de documents enfants.

Voir aussi hw_childrenobj et hw_getchilddoccollobj .

8.45.35 **hw_GetChildDocColl()** : Identifiant des documents fils d'un groupe

array **hw_getchilddoccoll** (int connection , int objectID)

hw_getchilddoccoll retourne un tableau avec les id des documents fils d'une collection.

Voir aussi hw_children et hw_getchildcoll .

8.45.36 **hw_GetChildDocCollObj()** : Attributs des documents fils d'un groupe

array **hw_getchilddoccollobj** (int connection , int objectID)

hw_getchilddoccollobj retourne un tableau contenant les attributs des documents fils du groupe objectID .

Voir aussi hw_childrenobj et hw_getchildcollobj .

8.45.37 **hw_GetObject()** : Lit les attributs d'un objet Hyperwave

mixed **hw_getobject** (int connection , mixed objectID , *string query*)

hw_GetObject retourne les attributs de l'objet dont l'identifiant est objectid , si le second paramètre est un entier. Si ce second paramètre est un tableau, la fonction retournera un tableau d'attributs. Dans ce cas, le dernier paramètre est aussi évalué.

query a la syntaxe suivante :

<expression> ::= "(" <expression> ")" |

"!" <expression> | /* NOT */

<expression> "||" <expression> | /* OR */

<expression> "&&" <expression> | /* AND */

<attribute> <operator> <value>

<attribute> ::= /* * n'importe quel attribut (Title, Author, DocumentType ...) */

<operator> ::= "=" | /* égal */

"<" | /* moins que (comparaison de type chaîne) */

">" | /* plus que (comparaison de type chaîne) */

"~" /* recherche par expression rationnelle */

query permet de sélectionner une nouvelle fois certains objets dans la liste des objets donnés. Contrairement aux autres requêtes, celle-ci peut utiliser des attributs non indexés. Le nombre d'attributs retournés dépend de la requête de la requête et des autorisations d'accès aux objets.

Voir aussi [hw_getandlock](#) et [hw_getobjectbyquery](#) .

8.45.38 hw_GetObjectByQuery() : Recherche un objet

array **hw_getobjectbyquery** (int connection , string query , int max_hits)

[hw_getobjectbyquery](#) recherche un objet sur tout le serveur et retourne un tableau d' object ids. Le nombre maximum d'objets est limité par max_hits . Si max_hits vaut -1, il n'y a pas de limite.

La requête ne fonctionnera qu'avec des attributs indexés.

Voir aussi [hw_getobjectbyqueryobj](#) .

8.45.39 hw_GetObjectByQueryColl() : Recherche un objet dans un groupe

array **hw_getobjectbyquerycoll** (int connection , int objectID , string query , int max_hits)

[hw_getobjectbyquerycoll](#) recherche un objet sur tout le groupe objectID et retourne un tableau d' object records. Le nombre maximum d'objet est limité par objectID . Si objectID vaut -1 il n'y a pas de limite.

La requête ne fonctionnera qu'avec des attributs indexés.

Voir aussi [hw_getobjectbyquerycollobj](#) .

8.45.40 **hw_GetObjectByQueryCollObj()** : Recherche un objet dans un groupe

array **hw_getobjectbyquerycollobj** (int connection , int objectID , string query , int max_hits)

hw_getobjectbyquerycollobj recherche un objet sur tout le groupe objectID et retourne un tableau d' object records. Le nombre maximum d'objets est limité par objectID . Si objectid vaut -1 il n'y a pas de limite.

La requête ne fonctionnera qu'avec des attributs indexés.

Voir aussi hw_getobjectbyquerycoll .

8.45.41 **hw_GetObjectByQueryObj()** : Recherche un objet

array **hw_getobjectbyqueryobj** (int connection , string query , int max_hits)

hw_getobjectbyqueryobj recherche un objet sur tout le serveur et retourne un tableau d' object records. Le nombre maximum d'objets est limité par max_hits . Si max_hits vaut -1, il n'y a pas de limite.

La requête ne fonctionnera qu'avec des attributs indexés.

Voir aussi hw_getobjectbyquery .

8.45.42 **hw_GetParents()** : Identifiant d'objet des parents

array **hw_getparents** (int connection , int objectID)

hw_getparents retourne un tableau indexé avec les identifiants des objets parents de objectID .

8.45.43 **hw_GetParentsObj()** : Attributs des parents

array **hw_getparentsobj** (int connection , int objectID)

hw_getparentsobj retourne un tableau indexé, avec les attributs et un tableau associé d'informations statistiques à propos des attributs. Ce tableau associé est le dernier élément du tableau retourné. Chaque attribut appartient au père de l'objet objectID .

8.45.44 **hw_getrellink()** : Crée le lien depuis la source vers le destinataire, relativement à rootid

string **hw_getrellink** (int link , int rootid , int sourceid , int destid)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.45.45 hw_GetRemote() : Retourne un document distant

int **hw_getremote** (int connection , int objectID)

hw_getremote retourne un document distant. Les documents distants sont, en Hyperwave, des documents lus depuis une source externe. La plupart des documents éloignés sont des pages web externes, ou des requêtes sur une base de données. Afin de pouvoir accéder à des sources externes, grâce aux documents distants, Hyperwave introduit l'interface HGI (Hyperwave Gateway Interface) qui est similaire à CGI. Actuellement, seuls les protocoles de FTP, HTTP et certaines bases de données sont accessibles avec HGI. hw_getremote retourne le document de la source distante. Si vous voulez utiliser cette fonction, il vous faut vous familiariser avec HGIs. Il est aussi préférable d'utiliser PHP plutôt que Hyperwave pour accéder aux sources externes. Le support des bases de données sera plus difficile avec Hyperwave que PHP.

Voir aussi hw_getremotechildren .

8.45.46 hw_getremotechildren() : Retourne les fils d'un document distant

mixed **hw_getremotechildren** (int connection , string object_record)

hw_getremotechildren retourne les fils d'un document distant. Les fils d'un document distant sont des documents distants eux mêmes. Cela est cohérent si une requête sur une base de données doit être rendue plus sélective, comme expliqué dans Hyperwave Programmers' Guide. Si le nombre de fils est 1, la fonction va retourner le document lui-même, formaté Hyperwave Gateway Interface (HGI). Si le nombre de fils est supérieur à 1, la fonction retournera un tableau d'attributs qui pourra servir à une nouvelle requête avec hw_getremotechildren . Ces attributs sont virtuels et n'existent pas sur le serveur Hyperwave, et ainsi, n'ont pas d'identifiant d'objet valide. L'ordre exact de ces objets est du ressort de HGI. Si vous voulez utiliser cette fonction, vous devez être très avec familier HGIs. Il vaut mieux PHP plutôt que Hyperwave pour accéder aux fichiers distants. Le support de base de données y est bien meilleur.

Voir aussi hw_getremote .

8.45.47 hw_GetSrcByDestObj() : Retourne les ancrages qui pointent sur un objet

array **hw_getsrcbydestobj** (int connection , int objectID)

hw_getsrcbydestobj retourne les attributs de tous les ancrages qui pointent sur objectID . L'objet peut être un document ou un autre ancrage, de type destination.

Voir aussi hw_getanchors .

8.45.48 hw_GetText() : Retourne un document texte

int **hw_gettext** (int connection , int objectID , mixed rootID/prefix)

hw_gettext retourne le document de l'objet objectID . Si le document possède des ancrages qui peuvent être insérés, ils le seront déjà. L'option rootID/prefix peut être une chaîne ou un entier. Si

c'est un entier, il détermine la méthode d'insertion des liens dans le document. Par défaut, il vaut 0 et les liens seront construits en fonction du nom de l'objet cible. Cela sert beaucoup dans les applications web. Si un lien pointe sur un objet avec le nom 'film_internet' le lien HTML sera . La position réelle de la source et de la cible dans la hiérarchie seront ignorés. Vous devrez modifier votre site web pour qu'il réécrive les URL, comme '/mon_script.php3/film_internet'. 'mon_script.php3' devra analyser \$PATH_INFO et savoir rechercher le document '/mon_script.php3/film_internet'. Si vous ne voulez pas de ce comportement, vous pouvez affecter à rootID/prefix n'importe quel préfixe. Dans ce cas, ce sera une chaîne.

Si rootID/prefix est un entier différent de 0 le lien sera construit avec tous les noms de la hiérarchie, en commençant à l'objet d'identifiant rootID/prefix, et séparé par des slash. Si, par exemple, le document 'film_internet' est situé à 'a-b-c-film_internet' et '-' qui sert de séparateur hiérarchique de niveau sur le serveur Hyperwave et le document source est situé dans 'a-b-d-source' alors, le lien HTML sera : . Cela est très pratique si vous voulez télécharger tout le contenu d'un serveur sur un disque, et faire une carte du système sur votre disque.

hw_gettext n'est opérationnelle qu'avec des documents de pur texte. Elle n'ouvrira pas de canal spécial de transfert et, ainsi, bloquera le script le temps du transfert.

Voir aussi hw_pipedocument, hw_free_document, hw_document_bodytag, hw_document_size et hw_output_document.

8.45.49 hw_getusername() : Nom de l'utilisateur actuellement identifié

string **hw_getusername** (int connection)

hw_getusername retourne le nom d'utilisateur utilisé par la connexion.

8.45.50 hw_Identify() : Identifie un utilisateur

string **hw_identify** (int link , string username , string password)

hw_identify identifie un utilisateur dont le nom d'utilisateur est username et le mot de passe password. L'identification n'est valide que pour la session en cours. Je ne pense pas que cette fonction serve souvent. Dans la plupart des cas, il est plus simple de s'identifier lors de l'ouverture de la connexion.

Voir aussi hw_connect.

8.45.51 hw_InCollections() : Vérifie qu'un identifiant d'objet est dans un groupe

array **hw_incollections** (int connection , array object_id_array , array collection_id_array , int return_collections)

hw_incollections vérifie qu'un ensemble d'objets (documents ou groupes) donnés par object_id_array fait partie des groupes listés par collection_id_array. Lorsque le quatrième paramètre return_collections vaut 0, le sous-ensemble d'identifiants qui font partie d'un groupe (i.e. les documents ou groupes qui sont fils d'un ou plusieurs groupe, ou leurs fils, récursivement) est

retourné sous la forme d'un tableau. Cette option permet de mettre en valeur la partie de l'arborescence qui contient le résultat d'une requête, dans un sens graphique.

8.45.52 **hw_Info()** : Informations à propos d'une connexion

string **hw_info** (int connection)

hw_info retourne les informations de la connexion courante. La chaîne retournée a le format suivant : <Serverstring>, <Host>, <Port>, <Username>, <Port of Client>, <Byte swapping>

8.45.53 **hw_Inscoll()** : Insère un groupe

int **hw_inscoll** (int connection , int objectID , array object_array)

hw_inscoll insère un nouveau groupe, avec les attributs object_array dans le groupe objectID .

8.45.54 **hw_InsdDoc()** : Insère un document

int **hw_insdDoc** (resource connection , int parentID , string object_record , *string text*)

hw_insdDoc insère un nouveau document avec les attributs object_record , dans le groupe parentID . Cette fonction insère soit un objet avec ses seuls attributs, soit un objet ascii, avec text s'il est fourni. Si vous voulez insérer un document de type général, utilisez plutôt hw_insertdocument .

Voir aussi hw_insertdocument et hw_inscoll .

8.45.55 **hw_insertanchors()** : Insère uniquement des ancrs dans du texte

bool **hw_insertanchors** (int hwdoc , array anchorecs , array dest , array urlprefixes)

Attention
 Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.45.56 **hw_InsertDocument()** : Insère un document dans un groupe

int **hw_insertdocument** (int connection , int parent_id , int hw_document)

hw_insertdocument insère un document dans le groupe parent_id . Le document doit avoir été créé auparavant avec hw_new_document . Assurez-vous que les attributs du nouveau document contiennent au moins les indications suivantes : Type, DocumentType, Title et Name. Vous aurez aussi parfois besoin de MimeType. La fonction retourne l'identifiant de l'objet inséré, ou bien FALSE .

Voir aussi hw_pipedocument .

8.45.57 `hw_InsertObject()` : Insère un objet record

int `hw_insertobject` (int connection , string object_rec , string parameter)

`hw_insertobject` insère un objet dans le serveur. L'objet peut être n'importe quel objet Hyperwave valide. Reportez-vous à la documentation HG-CSP pour plus de détails sur les paramètres.

Note : si vous voulez insérer une ancre, l'attribut Position doit être mis à la valeur start/end (début ou fin) ou encore 'invisible'. Les positions invisibles sont nécessaires si l'annotation n'a pas de liens correspondant dans le texte de l'annotation.

Voir aussi `hw_pipedocument` , `hw_insertdocument` , `hw_insdoc` et `hw_inscoll` .

8.45.58 `hw_mapid()` : Représente un id global en un id virtuel local

int `hw_mapid` (int connection , int server_id , int object_id)

`hw_mapid` représente l'id d'un objet global de n'importe quel serveur Hyperwave, même si vous ne vous y êtes pas connecté avec `hw_connect` , avec un id d'objet local virtuel. Cet id d'objet local peut alors être utilisé comme n'importe quel id d'objet : par exemple on peut obtenir l'enregistrement d'objet avec la fonction `hw_getobject` . L'id du serveur est la première partie de l'id global (GOid) de l'objet, qui est en fait une adresse IP.

Note : afin d'utiliser cette fonction, vous devez lever le flag `F_DISTRIBUTED`, ce qui ne peut être fait qu'à la compilation. Par défaut, il n'est pas levé. Lisez les commentaires de début du fichier `hg_comm.c`

8.45.59 `hw_Modifyobject()` : Modifie les attributs d'objet record

bool `hw_modifyobject` (int connection , int object_to_change , array remove , array add , int mode)

`hw_Modifyobject` permet d'effacer, d'ajouter ou de modifier les attributs d'un objet. L'objet est repéré par son identifiant `object_to_change` . Le premier tableau, `remove` , est la liste des attributs à effacer. Le deuxième tableau, `add` , est celle des attributs à ajouter. Afin de modifier un attribut, il vous faudra d'abord l'effacer, puis l'ajouter à nouveau. `hw_modifyobject` effacera toujours les attributs avant de les ajouter, à moins que la valeur de l'attribut à effacer ne soit pas une chaîne, ou un tableau.

Le dernier paramètre détermine si la modification est récursive ou pas. 1 signifie que la modification est récursive. Si un objet ne peut pas être modifié, il sera ignoré. `hw_error` n'indiquera alors pas toujours d'erreur, même si certains objets n'ont pas pu être modifiés.

Les clés des deux tableaux sont les noms des attributs. La valeur de chaque élément peut être un tableau, une chaîne ou n'importe quoi d'autre. Dans le cas du tableau, la valeur de l'attribut est construite en séparant chaque élément par un point virgule. Dans le cas de la chaîne, elle sert directement de valeur. Une chaîne vide provoquera un effacement de l'attribut. Si la valeur n'est ni un tableau, ni une chaîne, aucune opération ne sera effectuée. Cela est nécessaire si vous voulez ajouter un attribut complètement nouveau, pas seulement une nouvelle valeur pour un attribut existant. Si le tableau d'effacement contenait une chaîne vide comme attribut, le serveur tenterait d'effacer l'attribut, ce qui échouerait de toute manière, car cet attribut n'existe pas. L'ajout de cet attribut échouerait aussi. Affecter la valeur de 0 à cet attribut ne l'effacerait pas, et l'ajout

fonctionnerait.

Si vous voulez changer l'attribut 'Nom' de valeur courante 'livres' en 'articles' vous devrez faire deux tableaux, et appeler `hw_modifyobject` .

Modification d'un attribut

```
<?php
// $connect est une connexion valide
// $objid est l'identifiant de l'objet
$remarr = array("Name" => "books");
$addarr = array("Name" => "articles");
$hw_modifyobject($connect, $objid, $remarr, $addarr);
?>
```

Afin d'effacer/ajouter une paire nom=valeur aux attributs d'un objet, utilisez simplement les tableaux d'effacement et d'ajout, et laissez le dernier/troisième paramètre vide. Si l'attribut est le premier de ce nom à ajouter, donnez une valeur entière à cet élément.

Ajouter un nouvel attribut

```
<?php
// $connect est une connexion Hyperwave valide
// $objid est l'identifiant de l'objet à modifier
$remarr = array("Name" => 0);
$addarr = array("Name" => "articles");
$hw_modifyobject($connect, $objid, $remarr, $addarr);
?>
```

Note

Les attributs multilingues, tels que 'Title', peuvent être modifiés de deux façons : soit en fournissant la valeur de ces attributs de manière native (langue :valeur), soit en fournissant un tableau avec les éléments de chaque langue, comme décrit ci-dessus. L'exemple deviendrait alors :

Modifier l'attribut de Titre (Title)

```
<?php
$remarr = array("Title" => "en:Books");
$addarr = array("Title" => "en:Articles");
$hw_modifyobject($connect, $objid, $remarr, $addarr);
?>
```

ou

Modifier l'attribut Title

```
<?php
$remarr = array("Title" => array("en" => "Books"));
$addarr = array("Title" => array("en" => "Articles", "ge"=>"Artikel"));
$hw_modifyobject($connect, $objid, $remarr, $addarr);
?>
```

Pour supprimer l'entrée française 'Livres' et ajouter l'entrée 'Articles' et l'entrée allemande 'Artikel'.

Suppression d'un attribut

```
<?php
$remarr = array("Title" => "");
$addarr = array("Title" => "en:Articles");
$hw_modifyobject($connect, $objid, $remarr, $addarr);
?>
```

Note

Cet exemple va effacer tous les attributs avec le nom 'Title' et ajouter un nouvel attribut 'Title'. Cela peut être pratique pour effacer des attributs récursivement.

Note
Si vous devez effacer tous les attributs avec un certain nom, vous devez passer une chaîne vide comme valeur.
Note
Seuls les attributs 'Title', 'Description' et 'Keyword' gèrent correctement le préfixe de langue. Pour les autres attributs qui ne portent pas de préfixe de langage, le préfixe 'xx' sera assigné.
Note
L'attribut 'Name' est un peu particulier. Dans certains cas, il ne peut pas être complètement effacé. Vous aurez alors le message 'Change of base attribute' (l'apparition de cette erreur n'est pas très claire). Ainsi, vous aurez à ajouter une nouvelle entrée pour Name puis, effacer l'ancien.
Note
Il ne faut pas encadrer cette fonction par des appels à hw_getandlock et hw_unlock . hw_modifyobject le fait de manière interne.
Retourne TRUE si aucune erreur ne survient et FALSE sinon.

8.45.60 [hw_mv\(\)](#) : Déplace un objet

int [hw_mv](#) (int connection , array object_id_array , int source_id , int destination_id)

[hw_mv](#) déplace les objets dont les identifiants sont passés dans le tableau `source_id` , depuis le `source_id` dans le `destination_id` . Si `destination_id` vaut 0, les objets ne seront plus insérés dans le groupe (ni dans le serveur). Dans ce cas, si une instance était la dernière instance d'un objet, l'objet sera effacé. Si vous voulez effacer toutes les instances d'un coup, utilisez [hw_deleteobject](#) .

La valeur retournée est le nombre d'objet déplacés.

Voir aussi [hw_cp](#) et [hw_deleteobject](#) .

8.45.61 [hw_New_Document\(\)](#) : Crée un nouveau document

int [hw_new_document](#) (string object_record , string document_data , int document_size)

[hw_new_document](#) retourne un nouveau document Hyperwave avec comme données `document_data` et comme attributs `object_record` . La longueur de `document_data` doit être donnée dans `document_size` . Cette fonction n'insère pas l'objet dans le serveur Hyperwave.

Voir aussi [hw_free_document](#) , [hw_document_size](#) , [hw_document_bodytag](#) , [hw_output_document](#) et [hw_insertdocument](#) .

8.45.62 [hw_objrec2array\(\)](#) : Convertit les attributs d'un objet en tableau

array [hw_objrec2array](#) (string object_record , array format)

[hw_objrec2array](#) convertit les attributs `object_record` d'un objet en un tableau. Les clés du tableau seront les noms des attributs. Les attributs multiples comme 'Title', dans différentes langues, seront rassemblés dans un autre tableau. Une clé est la partie gauche d'un attribut. Cette partie doit être longue d'au moins deux caractères. Les autres valeurs multiples d'un attribut sans préfixe forment un tableau indexé. Si le paramètre optionnel `format` est omis, les attributs 'Title', 'Description' et 'Keyword' seront traités comme des attributs de langue, et les attributs 'Group', 'Parent' et 'HtmlAttr'

le seront comme des attributs non préfixés, à valeur multiples. En passant un tableau qui contient le type de chaque attribut, vous pouvez modifier ce comportement. Le tableau est un tableau associatif, avec les attributs comme nom d'index, et la valeur étant l'une des suivantes : HW_ATTR_LANG or HW_ATTR_NONE

Voir aussi [hw_array2objrec](#) .

8.45.63 **hw_Output_Document()** : Affiche hw_document

bool **hw_output_document** (int hw_document)

[hw_output_document](#) affiche hw_document sans la balise BODY.

Pour la compatibilité ascendante, [hw_outputdocument](#) est aussi accepté, mais il est déconseillé.

8.45.64 **hw_pConnect()** : Crée une connexion persistante

int **hw_pconnect** (string host , int port , *string username* , string password)

[hw_pConnect](#) retourne un index de connexion en cas de succès, et FALSE si la connexion n'a pas pu être créée. [hw_pconnect](#) ouvre une connexion persistante à un serveur Hyperwave. Tous les arguments doivent être entre guillemets, hormis le numéro de port port . Les arguments username et password sont optionnels, et peuvent être ignorés. Dans ce cas, aucune identification ne sera faite : connexion anonyme. Cette fonction retourne une ressource de connexion qui sera utilisée par les autres fonctions Hyperwave. Vous pouvez ouvrir plusieurs connexions persistantes simultanées.

Voir aussi [hw_connect](#) .

8.45.65 **hw_PipeDocument()** : Retourne un document

int **hw_pipedocument** (int connection , int objectID , *array url_prefixes*)

[hw_pipedocument](#) retourne le document Hyperwave d'objet id objectID . Si le document a des ancrages, ils seront insérés. Le document sera transmis via une connexion de données spéciale, qui ne bloque pas la connexion de contrôle (ie, le serveur n'attend pas la fin du transfert pour rendre la main).

Voir aussi [hw_gettext](#) , [hw_free_document](#) , [hw_document_size](#) , [hw_document_bodytag](#) et [hw_output_document](#) .

8.45.66 **hw_Root()** : Objet id de la racine

int **hw_root** ()

[hw_root](#) retourne l'objet ID de la racine. Actuellement, cet identifiant est toujours 0. L'ensemble des fils de la racine est celui du serveur courant.

8.45.67 **hw_setlinkroot()** : Configure l'identifiant vers lequel les liens sont calculés

int **hw_setlinkroot** (int link , int rootid)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.45.68 **hw_stat()** : Retourne le statut de la chaîne

string **hw_stat** (int link)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.45.69 **hw_Unlock()** : Déverrouille un objet

bool **hw_unlock** (int connection , int objectID)

hw_unlock déverrouille un document, et laisse l'accès aux autres utilisateurs.

Voir aussi hw_getandlock .

8.45.70 **hw_Who()** : Liste des utilisateurs actuellement identifiés

array **hw_who** (int connection)

hw_who retourne un tableau contenant la liste des utilisateurs actuellement connectés au serveur Hyperwave. Chaque élément du tableau est lui-même un tableau qui contient l'identifiant de l'élément, le nom, le système, la date de connexion (onSinceDate), l'heure de connexion (onSinceTime), la durée de connexion (TotalTime) et self. 'self' vaut 1 si cette entrée appartient à l'utilisateur qui a appelé la fonction.

8.46 Hyperwave API

8.46.1 Introduction

Hyperwave a été développé à l' IICM à Graz. Le projet a commencé avec le nom de HYPER-G , puis il est devenu Hyperwave, lors de sa commercialisation en 1996.

Hyperwave n'est pas un logiciel libre. Sa version courante, la 5.5, est disponible à <http://www.hyperwave.com/> . Une version limitée dans le temps est commandable gratuitement (30 jours).

Voir aussi le module Hyperwave .

Hyperwave est un système d'information comparable à une base de données (HIS , Hyperwave Information Server). Il se concentre sur le stockage et la gestion de documents. Un document peut être n'importe quelle pièce de données qui soit stockable dans un fichier. Chaque document est accompagné de son propre enregistrement. L'enregistrement contient les méta-données du document. Les méta-données sont des listes d'attributs qui peuvent être manipulés par l'utilisateur. Certains attributs sont gérés par le serveur Hyperwave, les autres peuvent être modifiés par l'utilisateur.

8.46.2 Pré-requis

Depuis 2001, il y a un SDK Hyperwave disponible. Il supporte Java, JavaScript et C++. Cette extension PHP est basée sur l'interface C++. Pour activer le support hwapi de PHP, vous devez installer le SDK d'abord.

8.46.3 Installation

Après avoir installé le SDK Hyperwave, compilez PHP avec l'option `--with-hwapi[=DIR]` .

8.46.4 intégration dans Apache

L'intégration dans Apache et éventuellement d'autres serveurs est déjà décrite dans le module Hyperwave , qui a été la première extension à se connecter au serveur Hyperwave.

8.46.5 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable	Historique
<code>hwapi.allow_persistent</code>	"0"	PHP_INI_SYSTEM	

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à Directives du `php.ini` .

8.46.6 Types de ressources

Cette extension ne définit aucune ressource.

8.46.7 Constantes pré-définies

Cette extension ne définit aucune constante.

8.46.8 Classes

L'API fournie avec l'extension HW_API est complètement orientée objet. Elle est très similaire à l'interface C++ du SDK Hyperwave. Elle est constituée des classes suivantes :

- HW_API
- HW_API_Object
- HW_API_Attribute
- HW_API_Error
- HW_API_Content
- HW_API_Reason

Certaines classes basiques comme HW_API_String , HW_API_String_Array , etc., qui existent dans le SDK Hyperwave SDK n'ont pas été implémentées, car PHP a de puissants ersatz pour elles.

Chaque classe dispose de certaines méthodes, dont les noms sont identiques à leur alter ego du SDK Hyperwave SDK. Le passage des arguments à ces fonctions diffère de toutes les autres extensions PHP, mais est proche de l'API C++ du SDK Hyperwave. Au lieu de passer plusieurs paramètres, ils sont tous placés dans un tableau associatif, et transmis sous la forme d'un seul paramètre. Les paramètres les plus communs sont listés ci-dessous. Si d'autres paramètres sont nécessaires, ils seront documentés à chaque cas.

- objectIdentifier Le nom ou l'identifiant de l'objet, e.g. "rootcollection", "0x873A87680x00000002".
- parentIdentifier Le nom ou l'identifiant de l'objet qui est considéré comme parent.
- object Une instance de la classe HW_API_Object.
- parameters Une instance de la classe HW_API_Object.
- version La version de l'objet.
- mode Un entier qui détermine la méthode d'exécution de l'opération.
- attributeSelector Un tableau de chaînes de caractères, chacun contenant le nom d'un attribut. C'est le cas si vous lisez des enregistrements d'objets, et que vous voulez inclure certains attributs.
- objectQuery Une requête pour sélectionner certains objets dans une liste. Elle est utilisée pour réduire le nombre d'objets qui a été livré par une fonction comme [hw_api->children](#) ou [hw_api->find](#) .

Note

Les méthodes retournant un booléen peuvent retourner TRUE , FALSE ou un objet HW_API_Error .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)

- [intégration dans Apache](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Classes](#)
- [hw_api attribute->key](#)
- [hw_api attribute->langdepvalue](#)
- [hw_api attribute->value](#)
- [hw_api attribute->values](#)
- [hw_api attribute](#)
- [hw_api->checkin](#)
- [hw_api->checkout](#)
- [hw_api->children](#)
- [hw_api content->mimetype](#)
- [hw_api content->read](#)
- [hw_api->content](#)
- [hw_api->copy](#)
- [hw_api->dbstat](#)
- [hw_api->dcstat](#)
- [hw_api->dstanchors](#)
- [hw_api->dstofsrcanchor](#)
- [hw_api error->count](#)
- [hw_api error->reason](#)
- [hw_api->find](#)
- [hw_api->ftstat](#)
- [hwapi hgscsp](#)
- [hw_api->hwstat](#)
- [hw_api->identify](#)
- [hw_api->info](#)
- [hw_api->insert](#)
- [hw_api->insertanchor](#)
- [hw_api->insertcollection](#)
- [hw_api->insertdocument](#)
- [hw_api->link](#)
- [hw_api->lock](#)
- [hw_api->move](#)
- [hw_api content](#)
- [hw_api object->assign](#)
- [hw_api object->attreditable](#)
- [hw_api object->count](#)
- [hw_api object->insert](#)
- [hw_api object](#)
- [hw_api object->remove](#)
- [hw_api object->title](#)
- [hw_api object->value](#)
- [hw_api->object](#)
- [hw_api->objectbyanchor](#)
- [hw_api->parents](#)
- [hw_api reason->description](#)
- [hw_api reason->type](#)
- [hw_api->remove](#)
- [hw_api->replace](#)
- [hw_api->setcommittedversion](#)
- [hw_api->srcanchors](#)

- [hw_api->srcsofdst](#)
- [hw_api->unlock](#)
- [hw_api->user](#)
- [hw_api->userlist](#)

8.46.10 [hw_api_attribute->langdepvalue\(\)](#) : Retourne la valeur d'un attribut dans une langue

string [hw_api_attribute->langdepvalue](#) (string language)

[hw_api_attribute->langdepvalue](#) retourne la valeur de l'attribut dans la langue language .

Voir aussi [hwapi_attribute_value](#) .

8.46.11 [hw_api_attribute->value\(\)](#) : Retourne la valeur d'un attribut

string [hw_api_attribute->value](#) (void)

[hw_api_attribute->value](#) retourne la valeur de l'attribut.

Voir aussi [hwapi_attribute_key](#) et [hwapi_attribute_values](#) .

8.46.12 [hw_api_attribute->values\(\)](#) : Retourne toutes les valeurs de l'attribut

array [hw_api_attribute->values](#) (void)

[hw_api_attribute->values](#) retourne toutes les valeurs de l'attribut sous la forme d'un tableau de chaînes.

Voir aussi [hwapi_attribute_value](#) .

8.46.13 [hw_api_attribute\(\)](#) : Crée une nouvelle instance de la classe [hw_api_attribute](#)

`HW_API_Attribute` [hw_api_attribute](#) (*string name* , *string value*)

[hw_api_attribute->attribute](#) crée une nouvelle instance de la classe [hw_api_attribute](#) avec le nom *name* et la valeur *value* .

8.46.14 [hw_api->checkin\(\)](#) : Archive un objet

bool [hw_api->checkin](#) (array parameter)

[hw_api->checkin](#) archive un objet ou une hiérarchie d'objets. Le paramètre obligatoire *parameter* contient un 'objectIdentifier' et les éléments optionnels 'version', 'comment', 'mode' et 'objectQuery'. 'version' spécifie la version de l'objet. Elle est constituée de versions majeure et mineure, séparées

par des virgules. Si une version n'existe pas, la version mineure est incrémentée. 'mode' peut être l'une des constantes suivantes :

HW_API_CHECKIN_NORMAL

Archive et valide l'objet. L'objet doit être un document.

HW_API_CHECKIN_RECURSIVE

Si l'objet à archiver est une collection, tous les fils seront vérifiés récursivement, s'ils sont des documents. Tenter d'archiver une collection générera une erreur.

HW_API_CHECKIN_FORCE_VERSION_CONTROL

Archive un objet même s'il ne passe pas sous le contrôle de versions.

HW_API_CHECKIN_REVERT_IF_NOT_CHANGED

Vérifie s'il existe une version différente de la dernière version. A moins que cela ne soit le cas, l'objet sera alors archivé.

HW_API_CHECKIN_KEEP_TIME_MODIFIED

Conserve la date de modification de l'objet le plus récent.

HW_API_CHECKIN_NO_AUTO_COMMIT

L'objet n'est pas automatiquement validé lors de l'archivage.

Voir aussi [hwapi_checkout](#) .

8.46.15 hw_api->checkout() : Extrait un objet

bool **hw_api->checkout** (array parameter)

hw_api->checkout extrait un objet ou toute une hiérarchie d'objets. Le paramètre parameter contient l'élément obligatoire 'objectIdentifier' et les éléments optionnels 'version', 'mode' et 'objectQuery'. 'mode' peut prendre l'une des valeurs suivantes :

HW_API_CHECKIN_NORMAL

Extrait un objet. L'objet doit être un document.

HW_API_CHECKIN_RECURSIVE

Si l'objet à extraire est une collection, tous les fils seront extraits récursivement si ce sont des documents. Extraire une collection générera une erreur.

Voir aussi [hwapi_checkin](#) .

8.46.16 hw_api->children() : Retourne le fils d'un objet

array **hw_api->children** (array parameter)

[hw_api->children](#) retrouve le fils d'une collection ou l'attribut d'un document. Le fils peut être ensuite filtré en spécifiant une requête d'objet. Le paramètre parameter contient les éléments requis 'objectIdentifier' et les éléments optionnels 'attributeSelector' et 'objectQuery'.

La valeur retournée est un tableau d'objets de type HW_API_Object ou HW_API_Error .

Voir aussi [hwapi_parents](#) .

8.46.17 `hw_api_content->mimetype()` : Retourne le type MIME

string `hw_api_content->mimetype` (void)

`hw_api_content->mimetype` retourne le type MIME du contenu.

8.46.18 `hw_api_content->read()` : Lit le contenu

string `hw_api_content->read` (string `buffer` , int `len`)

`hw_api_content->read` lit `len` octets dans le buffer `buffer` .

8.46.19 `hw_api->content()` : Retourne le contenu d'un objet

HW_API_Content `hw_api->content` (array `parameter`)

`hw_api->content` retourne le contenu d'un document sous la forme d'un objet de type `hw_api_content` . Le paramètre `parameter` est un tableau qui contient l'élément obligatoire 'objectIdentifier' et l'élément optionnel 'mode'. Le mode peut être l'une des constantes suivantes : `HW_API_CONTENT_ALLLINKS` , `HW_API_CONTENT_REACHABLELINKS` ou `HW_API_CONTENT_PLAIN` . `HW_API_CONTENT_ALLLINKS` signifie qu'il faut insérer toutes les ancrs même si la destination n'est pas accessible. `HW_API_CONTENT_REACHABLELINKS` indique à `hw_api_content` qu'il faut insérer uniquement les destinations accessibles, et `HW_API_CONTENT_PLAIN` conduit à un document sans aucun lien.

8.46.20 `hw_api->copy()` : Copie physique

`hw_api_object hw_api->copy` (array `parameter`)

`hw_api->copy` fait une copie physique incluant le contenu éventuel, et retourne un nouvel objet ou un objet d'erreur. Le paramètre `parameter` est un tableau qui contient les éléments obligatoires 'objectIdentifier' et 'destinationParentIdentifier' et l'élément optionnel 'attributeSelector'.

Voir aussi [hwapi_move](#) et [hwapi_link](#) .

8.46.21 `hw_api->dbstat()` : Retourne des statistiques sur le serveur de bases

`hw_api_object hw_api->dbstat` (array `parameter`)

`dbstat` retourne des statistiques sur le serveur de bases.

Voir aussi [hwapi_dcstat](#) , [hwapi_hwstat](#) et [hwapi_ftstat](#) .

8.46.22 `hw_api->dcstat()` : Retourne des statistiques sur le serveur de cache de document

`hw_api_object hw_api->dcstat` (array parameter)

`hw_api->dcstat` retourne des statistiques sur le serveur de cache de document.

Voir aussi `hwapi_hwstat` , `hwapi_dbstat` et `hwapi_ftstat` .

8.46.23 `hw_api->dstanchors()` : Liste toutes les ancrs de destination

array `hw_api->dstanchors` (array parameter)

`hw_api->dstanchors` liste toutes les ancrs de destination. Le paramètre `parameter` contient l'élément obligatoire `'objectIdentifier'` et les éléments optionnels `'attributeSelector'` et `'objectQuery'`.

Voir aussi `hwapi_srcanchors` .

8.46.24 `hw_api->dstofsrcanchor()` : Retourne la destination d'une ancre source

`hw_api_object hw_api->dstofsrcanchor` (array parameter)

`hw_api->dstofsrcanchor` retourne l'objet de destination sur lequel pointe l'ancre source spécifiée. L'objet de destination peut être une ancre de destination ou un document entier. Le paramètre `parameter` contient l'élément obligatoire `'objectIdentifier'` et le paramètre optionnel `'attributeSelector'`.

Voir aussi `hwapi_srcanchors` , `hwapi_dstanchors` et `hwapi_objectbyanchor` .

8.46.25 `hw_api_error->count()` : Retourne le nombre de raisons d'erreurs

int `hw_api_error->count` (void)

`hw_api_error->count` retourne le nombre de raisons d'erreurs.

Voir aussi `hwapi_error_reason` .

8.46.26 `hw_api_error->reason()` : Retourne la première raison d'erreur

HW_API_Reason `hw_api_error->reason` (void)

`hw_api_error->reason` retourne la première raison d'erreur.

Voir aussi `hwapi_error_count` .

8.46.27 `hw_api->find()` : Recherche des objets

array `hw_api->find` (array parameter)

`hw_api->find` recherche des objets grâce à leur clé d'exécution et/ou grâce à une requête en texte plein. Les objets trouvés peuvent alors être filtrés à nouveau par une requête optionnelle. Les objets

sont triés par importance. La seconde opération de recherche est relativement lente, et ses résultats peuvent être limités à certaines valeurs. Cela permet de réaliser des recherches incrémentales, chacune retournant un sous-ensemble de tous les documents trouvés, en partant d'un index donné. Le paramètre `parameter` est un tableau qui contient les éléments obligatoires 'objectquery', 'scope', 'languages' et 'attributeselector'. En cas de recherche incrémentale, les éléments optionnels 'startIndex', 'numberOfObjectsToGet' et 'exactMatchUnit' peuvent aussi être passés.

8.46.28 `hw_api->ftstat()` : Retourne des statistiques sur le serveur de textes

`hw_api_object hw_api->ftstat (array parameter)`

`ftstat` retourne des statistiques sur le serveur de textes.

Voir aussi [hwapi_dcstat](#) , [hwapi_dbstat](#) et [hwapi_hwstat](#) .

8.46.29 `hwapi_hgcsp()` : Retourne un objet de classe `hw_api`

`HW_API hwapi_hgcsp (string hostname , int port)`

`hwapi_hgcsp` ouvre une connexion au serveur Hyperwave sur l'hôte `hostname` . Le protocole utilisé est HGCSPP. Si vous l'omettez, le port 418 sera utilisé.

Voir aussi [hwapi_hwtp](#) .

8.46.30 `hw_api->hwstat()` : Retourne les statistiques sur le serveur Hyperwave

`hw_api_object hw_api->hwstat (array parameter)`

`hwstat` retourne les statistiques sur le serveur Hyperwave.

Voir aussi [hwapi_dcstat](#) , [hwapi_dbstat](#) et [hwapi_ftstat](#) .

8.46.31 `hw_api->identify()` : S'identifie auprès du serveur Hyperwave

`bool hw_api->identify (array parameter)`

`hw_api->identify` s'identifie auprès du serveur Hyperwave. Le paramètre `parameter` est un tableau qui contient les éléments obligatoires 'username' et 'password'.

La valeur retournée sera un objet du type `HW_API_Error` si l'identification a échoué, ou bien `TRUE` si cela a fonctionné.

8.46.32 `hw_api->info()` : Retourne des informations sur la configuration du serveur

array `hw_api->info` (array parameter)

`hw_api->info` retourne des informations sur la configuration du serveur.

Voir aussi [hwapi_dcstat](#) , [hwapi_dbstat](#) , [hwapi_ftstat](#) et [hwapi_hwstat](#) .

8.46.33 `hw_api->insert()` : Insère un nouvel objet

hw_api_object `hw_api->insert` (array parameter)

`hw_api->insert` insère un nouvel objet. Le type d'objet peut être user, group, document ou anchor. Suivant le type de l'autre objet, des attributs supplémentaires doivent être assignés. Le paramètre parameter est un tableau qui contient les éléments obligatoires 'object' et 'content' (si l'objet est un document) et les éléments optionnels 'parameters', 'mode' et 'attributeSelector'. 'object' doit contenir tous les attributs de l'objet. 'parameters' est un objet qui contient les attributs supplémentaires, comme ceux de destination (attribut 'Parent'). 'content' est une combinaison des constantes suivantes :

HW_API_INSERT_NORMAL

L'objet est inséré sur le serveur.

HW_API_INSERT_FORCE_VERSION_CONTROL

HW_API_INSERT_AUTOMATIC_CHECKOUT

HW_API_INSERT_PLAIN

HW_API_INSERT_KEEP_TIME_MODIFIED

HW_API_INSERT_DELAY_INDEXING

Voir aussi [hwapi_replace](#) .

8.46.34 `hw_api->insertanchor()` : Insère un nouvel objet de type ancre

hw_api_object `hw_api->insertanchor` (array parameter)

`hw_api->insertanchor` est un raccourci pour [hwapi_insert](#) . Elle insère un objet de type ancre ('anchor') et assigne les attributs nécessaires. Le paramètre parameter contient les éléments obligatoires 'object' et 'documentIdentifier' et les éléments optionnels 'destinationIdentifier', 'parameter', 'hint' et 'attributeSelector'. L'élément 'documentIdentifier' spécifie le document d'insertion de l'ancre. La cible de l'ancre est spécifiée par 'destinationIdentifier' s'il existe. Si la cible n'existe pas, l'élément 'hint' doit être configuré avec le nom de l'objet qui est prévu pour être inséré ultérieurement. Une fois que l'insertion a été faite, la cible sera résolue automatiquement.

Voir aussi [hwapi_insertdocument](#) , [hwapi_insertcollection](#) et [hwapi_insert](#) .

8.46.35 `hw_api->insertcollection()` : Insère un nouvel objet de type collection

hw_api_object `hw_api->insertcollection` (array parameter)

[hw_api->insertcollection](#) est un raccourci pour la fonction [hwapi_insert](#) . Elle insère un objet de type collection et lui assigne certains attributs obligatoires. `parameter` est un tableau qui contient les éléments obligatoires 'object' et 'parentIdentifieur', et les éléments optionnels 'parameter' et 'attributeSelector'. Voyez [hwapi_insert](#) pour connaître le détail de chaque élément.

Voir aussi [hwapi_insertdocument](#) , [hwapi_insertanchor](#) et [hwapi_insert](#) .

8.46.36 [hw_api->insertdocument\(\)](#) : Insère un nouvel objet de type document

`hw_api_object hw_api->insertdocument (array parameter)`

[hw_api->insertdocument](#) est un raccourci pour [hwapi_insert](#) . Il insère un objet avec un contenu et assigne certains attributs obligatoires. Le paramètre `parameter` est un tableau qui contient les éléments obligatoires 'object', 'parentIdentifieur' et 'content' et les éléments optionnels 'mode', 'parameter' et 'attributeSelector'. Voyez [hwapi_insert](#) pour la signification de chacun des éléments.

Voir aussi [hwapi_insert](#) , [hwapi_insertanchor](#) et [hwapi_insertcollection](#) .

8.46.37 [hw_api->link\(\)](#) : Crée un lien avec un objet

`bool hw_api->link (array parameter)`

[hw_api->link](#) crée un lien avec un objet. Accéder à ce lien revient à accéder à l'objet vers lequel il pointe. Le paramètre `parameter` contient les éléments obligatoires de 'objectIdentifieur' et 'destinationParentIdentifieur'. 'destinationParentIdentifieur' est la collection de destination.

[hw_api->link](#) retourne TRUE en cas de succès et, sinon, un objet d'erreur.

Voir aussi [hwapi_copy](#) .

8.46.38 [hw_api->lock\(\)](#) : Verrouille un objet

`bool hw_api->lock (array parameter)`

[hw_api->lock](#) verrouille un objet pour le faire éditer exclusivement par l'utilisateur qui appelle cette fonction. L'objet ne peut être déverrouillé que par cet utilisateur ou par le système. Le paramètre `parameter` est un tableau qui contient l'élément obligatoire 'objectIdentifieur' et les paramètres optionnels 'mode' et 'objectquery'. 'mode' détermine comment l'objet est verrouillé.

HW_API_LOCK_NORMAL signifie que l'objet est verrouillé jusqu'à son déverrouillage.

HW_API_LOCK_RECURSIVE est uniquement valide pour une collection et verrouille tous les objets de la collection et des sous-collections. HW_API_LOCK_SESSION signifie que l'objet est verrouillé aussi longtemps que tient la session en cours.

Voir aussi [hwapi_unlock](#) .

8.46.39 `hw_api->move()` : Déplace un objet d'une collection à l'autre

`bool hw_api->move (array parameter)`

`hw_api->move` déplace un objet d'une collection à l'autre.

Voir aussi `hw_objrec2array` .

8.46.40 `hw_api_content()` : Crée une nouvelle instance de la classe `hw_api_content`

`HW_API_Content hw_api_content (string content , string mimetype)`

`hw_api_content` crée un nouvel objet de contenu à partir de la chaîne `content` . Le type MIME est défini à `mimetype` .

8.46.41 `hw_api_object->assign()` : Clone un objet

`bool hw_api_object->assign (array parameter)`

`hw_api_object->assign` clone les attributs d'un objet.

8.46.42 `hw_api_object->attreditable()` : Vérifie si un attribut est éditable

`bool hw_api_object->attreditable (array parameter)`

`hw_api_object->attreditable` vérifie si un attribut est éditable.

8.46.43 `hw_api_object->count()` : Retourne le nombre d'attributs

`int hw_api_object->count (array parameter)`

`hw_api_object->count` retourne le nombre d'attributs.

8.46.44 `hw_api_object->insert()` : Insère un nouvel attribut HyperWave

`bool hw_api_object->insert (HW_API_Attribute attribute)`

`hw_api_object->insert` ajoute l'attribut `attribute` à l'objet courant. `hw_api_object->insert` retourne TRUE en cas de succès et, sinon, FALSE .

Voir aussi `hwapi_object_remove` .

8.46.45 `hw_api_object()` : Crée une nouvelle instance de la classe `hw_api_object`

`hw_api_object` **hw_api_object** (array parameter)

`hw_api_object` crée une nouvelle instance de la classe `hw_api_object` .

Voir aussi `hwapi_lock` .

8.46.46 `hw_api_object->remove()` : Supprime un attribut HyperWave

bool `hw_api_object->remove` (string name)

`hw_api_object->remove` supprime l'attribut de nom `name` dans l'objet courant.

`hw_api_object->remove` retourne TRUE en cas de succès, et FALSE sinon.

Voir aussi `hwapi_object_insert` .

8.46.47 `hw_api_object->title()` : Retourne le titre de l'attribut

string `hw_api_object->title` (array parameter)

8.46.48 `hw_api_object->value()` : Retourne la valeur de l'attribut

string `hw_api_object->value` (string name)

`hw_api_object->value` retourne la valeur de l'attribut `name` , ou bien FALSE si une erreur est survenue.

8.46.49 `hw_api->object()` : Lit les informations d'attribut

`hw_api_object` **hw_api->object** (array parameter)

`hw_api->object` lit les informations d'attribut d'un objet de n'importe quelle version. Elle ne retournera pas le contenu du document. Le paramètre `parameter` est un tableau qui contient l'élément obligatoire 'objectIdentifier' et les éléments optionnels 'attributeSelector' et 'version'.

L'objet retourné est une instance de la classe `HW_API_Object` en cas de succès ou `HW_API_Error` en cas d'erreur.

Cet exemple simple lit un objet et vérifie les erreurs.

```
Lit un objet Hyperwave et le vérifie
<?php
function handle_error($error)
{
    $reason = $error->reason(0);
    echo 'Type: <strong>';
    switch ($reason->type()) {
        case 0:
            echo 'Erreur';
            break;
        case 1:
            echo 'Alerte';
    }
}
```

```

        break;
    case 2:
        echo 'Message';
        break;
    }
    echo "</strong><br />\n";
    echo "Description : " . $reason->description("en") . "<br />\n";
}

function list_attr($obj)
{
    echo "<table>\n";
    $count = $obj->count();
    for ($i=0; $i<$count; $i++) {
        $attr = $obj->attribute($i);
        printf("<tr><td><strong>%s</strong></td><td>%s</td></tr>\n",
            $attr->key(), $attr->value());
    }
    echo "</table>\n";
}

$hwapl = hwapi_hgcsp($g_config[HOSTNAME]);
$params = array("objectIdentifier" => "rootcollection", "attributeSelector"=>array("Title", "Name"));
$root = $hwapl->object($params);
if (get_class($root) == "HW_API_Error") {
    handle_error($root);
    exit;
}
list_attr($root);
?>

```

Voir aussi [hwapi_content](#) .

8.46.50 hw_api->objectbyanchor() : Retourne l'objet propriétaire d'une ancre

hw_api_object **hw_api->objectbyanchor** (array parameter)

[hw_api->objectbyanchor](#) retourne l'objet qui contient une ancre donnée. Le paramètre parameter est un tableau qui contient l'élément obligatoire 'objectIdentifier' et l'élément optionnel 'attributeSelector'.

Voir aussi [hwapi_dstofsrcanchor](#) , [hwapi_srcanchors](#) et [hwapi_dstanchors](#) .

8.46.51 hw_api->parents() : Retourne le parent d'un objet

array **hw_api->parents** (array parameter)

[hw_api->parents](#) lit le parent d'un objet. Le parent peut être filtré en spécifiant un objet de requête. Le paramètre parameter est un tableau qui contient l'élément obligatoire 'objectIdentifier' et les éléments optionnels 'attributeselector' et 'objectquery'.

La valeur retournée est un tableau d'objets de type HW_API_Object ou HW_API_Error .

Voir aussi [hwapi_children](#) .

8.46.52 `hw_api_reason->description()` : Retourne la description d'une raison

string `hw_api_reason->description` (void)

`hw_api_reason->description` retourne la description d'une raison.

8.46.53 `hw_api_reason->type()` : Retourne le type de raison

HW_API_Reason `hw_api_reason->type` (void)

`hw_api_reason->type` retourne le type de raison.

8.46.54 `hw_api->remove()` : Efface un objet

bool `hw_api->remove` (array parameter)

`hw_api->remove` efface un objet avec un parent spécifique. Les collections seront effacées récursivement. Vous pouvez passer un objet optionnel de requête pour filtrer les objets qui seront effacés. Un objet sera effacé physiquement si c'est sa dernière instance. Le paramètre `parameter` est un tableau qui contient les éléments obligatoires 'objectidentifiant' et 'parentidentifiant'. Si vous souhaitez effacer un utilisateur ou un groupe, 'parentidentifiant' peut être omis. L'élément optionnel 'mode' détermine comment l'effacement est réalisé. En mode normal, l'objet sera effacé physiquement lorsque toutes ses instances seront supprimées. En mode physique, toutes les instances d'un objet seront supprimées immédiatement. En mode `removelinks` toutes les références vers et depuis l'objet seront aussi effacées. En mode `nonrecursive`, l'effacement n'est pas récursif. Effacer une collection qui n'est pas vide causera une erreur.

Voir aussi [hwapi_move](#) .

8.46.55 `hw_api->replace()` : Remplace un objet

hw_api_object `hw_api->replace` (array parameter)

`hw_api->replace` remplace les attributs et le contenu d'un objet. Le paramètre `parameter` est un tableau qui contient les éléments obligatoires 'objectIdentifiant' et 'object', puis les paramètres optionnels 'content', 'parameters', 'mode' et 'attributeSelector'. 'objectIdentifiant' contient l'objet qui doit être remplacé. 'object' contient le nouvel objet. 'content' contient le nouveau contenu. 'parameters' contient des informations supplémentaires pour les documents HTML. `HTML_Language` est l'abréviation de la langue pour le titre. `HTML_Base` est l'attribut de base pour le contenu du document HTML. 'mode' peut être la combinaison de l'une des constantes suivantes :

`HW_API_REPLACE_NORMAL`

L'objet sur le serveur est remplacé par l'objet passé.

`HW_API_REPLACE_FORCE_VERSION_CONTROL`

`HW_API_REPLACE_AUTOMATIC_CHECKOUT`

`HW_API_REPLACE_AUTOMATIC_CHECKIN`

`HW_API_REPLACE_PLAIN`

`HW_API_REPLACE_REVERT_IF_NOT_CHANGED`

`HW_API_REPLACE_KEEP_TIME_MODIFIED`

Voir aussi [hwapi_insert](#) .

8.46.56 hw_api->setcommittedversion() : Valide une version autre que la dernière

hw_api_object **hw_api->setcommittedversion** (array parameter)

[hw_api->setcommittedversion](#) valide une version du document. La version validée est celle qui est visible par les utilisateurs ayant un accès en lecture. Par défaut, la dernière version est celle qui est validée.

Voir aussi [hwapi_checkin](#) , [hwapi_checkout](#) et [hwapi_revert](#) .

8.46.57 hw_api->srcanchors() : Retourne la liste de toutes les ancrs

array **hw_api->srcanchors** (array parameter)

[hw_api->srcanchors](#) lit toutes les ancrs sources d'un objet. Le paramètre parameter est un tableau qui contient l'élément obligatoire 'objectIdentifieur' et les éléments optionnels 'attributeSelector' et 'objectQuery'.

Voir aussi [hwapi_dstanchors](#) .

8.46.58 hw_api->srcsofdst() : Retourne les sources d'un objet de destination

array **hw_api->srcsofdst** (array parameter)

[hw_api->srcsofdst](#) retourne les sources d'un objet de destination. L'objet de destination peut être une ancre ou un document complet. Le paramètre parameter est un tableau qui contient l'élément obligatoire 'objectIdentifieur' et les éléments optionnels 'attributeSelector' et 'objectQuery'. La fonction retourne un tableau d'objet ou une erreur.

Voir aussi [hwapi_dstofsrcanchor](#) .

8.46.59 hw_api->unlock() : Déverrouille un objet HyperWave verrouillé

bool **hw_api->unlock** (array parameter)

[hw_api->unlock](#) déverrouille un objet verrouillé. Seuls l'utilisateur qui a verrouillé un objet et l'administrateur système peuvent le déverrouiller. Le paramètre parameter est un tableau qui contient les éléments obligatoires de objectIdentifieur et les paramètres optionnels de mode et objectquery . La signification de mode est la même que dans la fonction [hwapi_lock](#) .

[hw_api->unlock](#) retourne TRUE en cas de succès et, sinon, un objet d'erreur HW_API_Error.

Voir aussi [hwapi_lock](#) .

8.46.60 hw_api->user() : Retourne le propriétaire d'un objet

hw_api_object **hw_api->user** (array parameter)

hw_api->user retourne le propriétaire d'un objet.

Voir aussi hwapi_userlist .

8.46.61 hw_api->userlist() : Retourne la liste des utilisateurs connectés

array **hw_api->userlist** (array parameter)

hw_api->userlist retourne la liste des utilisateurs connectés.

Voir aussi hwapi_user .

8.47 Fonctions Firebird/InterBase

8.47.1 Introduction

Firebird/InterBase est une base de données relationnelles offrant la plus part des fonctionnalités décrites dans la norme ANSI SQL-92, qui fonctionne sous environnements Linux, Windows, et la plus part des systèmes Unix. Firebird/InterBase offre une excellente simultanéité, de hautes performances et un langage efficace pour l'écriture des procédures stockées et des triggers. Il est utilisé sur des systèmes de production depuis 1981.

Interbase est le nom de la variante commerciale de cette base de données créée par Borland/Inprise. Pour plus d'informations sur Interbase, allez à <http://www.borland.com/interbase/>.

Firebird est un projet commercialement indépendant de programmeurs C et C++, conseillés techniques, supportant le développement et assurant la compatibilité multi-plate-forme de la base de données relationnelle basé sur le code source offert par Inprise Corp (maintenant connu sous le nom de Borland Software Corp) sous la license InterBase Public License v.1.0 le 25 Juillet 2000. Pour plus d'informations sur Firebird, allez à <http://www.firebirdsql.org/>.

Note

Cette extension supporte Interbase version 5 et suivante ainsi que toutes les versions de Firbird. Le support d'Interbase version 5.x sera supprimé dans PHP 5.

Cette base de données utilise les guillemets simples (') pour échapper les caractères, un peu comme le fait Sybase. Ajoutez à votre fichier php.ini la directive suivante :

Configuration InterBase

```
magic_quotes_sybase = On
```

8.47.2 Pré-requis

8.47.3 Installation

Pour activer le support de ibase, vous devez compiler PHP avec l'option `--with-interbase[=DIR]`.

Note

Note aux utilisateurs Win32

Afin d'activer ce module dans l'environnement Windows, vous devez copier les bibliothèques gds32.dll depuis le dossier DLL de PHP/Win32 dans le dossier système SYSTEM32 de votre machine (par exemple : C:\WINNT\SYSTEM32 ou C:\WINDOWS\SYSTEM32). Dans le cas où vous auriez installé le serveur de base de données InterBase sur la même machine qui exploite PHP, vous aurez déjà ces DLL de prêts. Par conséquent, vous n'avez pas à copier la bibliothèque gds32.dll.

8.47.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini.

Nom	Par défaut	Modifiable	Historique
-----	------------	------------	------------

ibase.allow_persistent	"1"	PHP_INI_SYSTEM	
ibase.max_persistent	"-1"	PHP_INI_SYSTEM	
ibase.max_links	"-1"	PHP_INI_SYSTEM	
ibase.default_db	NULL	PHP_INI_SYSTEM	Disponible depuis PHP 5.0.0.
ibase.default_user	NULL	PHP_INI_ALL	
ibase.default_password	NULL	PHP_INI_ALL	
ibase.default_charset	NULL	PHP_INI_ALL	Disponible depuis PHP 5.0.0.
ibase.timestampformat	"%Y-%m-%d %H:%M:%S"	PHP_INI_ALL	
ibase.dateformat	"%Y-%m-%d"	PHP_INI_ALL	
ibase.timeformat	"%H:%M:%S"	PHP_INI_ALL	

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [Directives du php.ini](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

ibase.allow_persistent booléen

Accepte ou non les connexions persistantes à Firebird/Interbase.

ibase.max_persistent entier

Le nombre maximal de connexions persistantes Firebird/Interbase par processus. Les nouvelles connexions avec `ibase_pconnect()` ne seront pas persistantes si ce nombre maximal est atteint.

ibase.max_links entier

Le nombre maximal de connexions Firebird/Interbase par processus, incluant les connexions persistantes.

ibase.default_db chaîne de caractères

Le nom de la base de données par défaut lorsque `ibase_[p]connect()` est appelé sans base de données spécifique. Si cette valeur est définie et que le mode sécurisé (safe mode) SQL est activé, aucune connexion à d'autres bases de données que celle-ci ne sera autorisé.

ibase.default_user chaîne de caractères

Le nom d'utilisateur utilisé lors de la connexion à la base de données lorsqu'aucun n'est spécifié.

ibase.default_password chaîne de caractères

Le mot de passe utilisé lors de la connexion à la base de données lorsqu'aucun n'est spécifié.

ibase.default_charset chaîne de caractères

Le jeu de caractères utilisé lors de la connexion à la base de données lorsqu'aucun n'est spécifié.

ibase.timestampformat chaîne de caractères

ibase.dateformat chaîne de caractères

ibase.timeformat chaîne de caractères

Ces directives sont utilisées pour définir les formats de dates et d'heures qui seront utilisés lorsque des dates/heures seront retournées d'un jeux de résultat, ou lors de traitement d'arguments en paramètres dates/heures.

8.47.5 Types de ressources

8.47.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Les constantes suivantes peuvent être passées à la fonction `ibase_trans` pour spécifier le comportement du traitement.

Constante	Description
IBASE_DEFAULT	Définit le comportement par défaut. Ce comportement est déterminé par la bibliothèque cliente, qui est définie comme <code>IBASE_WRITE IBASE_CONCURRENCY IBASE_WAIT</code> dans la plus part des cas.
IBASE_READ	Démarre une transaction en lecture seul
IBASE_WRITE	Démarre une transaction en lecture / écriture
IBASE_CONSISTENCY	Démarre une transaction avec le niveau d'isolation définie à 'consistency', ce qui signifie que la transaction courante ne peut lire depuis des tables qui ont été modifiées par d'autres transactions.
IBASE_CONCURRENCY	Démarre une transaction avec le niveau d'isolation définie à 'concurrency' (ou 'snapshot'), ce qui signifie que la transaction peut accéder à toutes les tables, mais ne peut pas voir les modifications apportées par d'autres transactions une fois que la transaction a débuté.
IBASE_COMMITTED	Démarre une transaction avec le niveau d'isolation définie à 'read committed'. Ce flag doit être associé avec soit la constante <code>IBASE_REC_VERSION</code> , soit la constante <code>IBASE_REC_NO_VERSION</code> . Ce niveau d'isolation vous permet d'accéder aux modifications effectuées après le début de la transaction. Si la constante <code>IBASE_REC_NO_VERSION</code> est spécifiée, seulement la dernière version des lignes pourra être lue. Si la constante <code>IBASE_REC_VERSION</code> est spécifiée, une ligne peut toujours être lue lorsqu'une modification est en attente dans une transaction concurrente.
IBASE_WAIT	Indique que la transaction peut attendre, puis réessayer lorsqu'un conflit apparaît.
IBASE_NOWAIT	Indique que la transaction échouera immédiatement lorsqu'un conflit apparaît.

Les constantes suivantes peuvent être passées aux fonctions `ibase_fetch_row`, `ibase_fetch_assoc` ou `ibase_fetch_object` pour spécifier leurs comportements.

Constante	Description
IBASE_FETCH_BLOBS	Aussi disponible sous le nom <code>IBASE_TEXT</code> pour des raisons de compatibilité ascendante. Permet de lire le contenu d'un BLOB 'inline' au lieu de le parcourir en utilisant un identifiant de BLOB.
IBASE_FETCH_ARRAYS	Permet de lire un tableau 'inline'. Sinon, les identifiants de tableaux sont retournés. Les identifiants de tableaux ne peuvent être passés uniquement comme arguments aux requêtes INSERT, car aucune fonction pour traiter les identifiants de tableaux n'est actuellement disponible.
IBASE_UNIXTIME	Permet de retourner les champs date et heure non pas comme des chaînes de caractères mais comme des timestamps UNIX (le nombre de secondes depuis l'époque UNIX, qui est le 1-Jan-1970 0:00 UTC). Cela peut être problématique si vous utilisez des dates antérieures à 1970 sur

quelques systèmes.

Les constantes suivantes sont utilisées pour passer des requêtes et des options à l'API (ibase_server_info , ibase_db_info , ibase_backup , ibase_restore et ibase_maintain_db). Merci de consulter le manuel Firebird/InterBase pour plus d'informations sur la significations de ces options.

```

IBASE_BKP_IGNORE_CHECKSUMS
IBASE_BKP_IGNORE_LIMBO
IBASE_BKP_METADATA_ONLY
IBASE_BKP_NO_GARBAGE_COLLECT
IBASE_BKP_OLD_DESCRIPTIONS
IBASE_BKP_NON_TRANSPORTABLE
IBASE_BKP_CONVERT
    Options de ibase_backup
IBASE_RES_DEACTIVATE_IDX
IBASE_RES_NO_SHADOW
IBASE_RES_NO_VALIDITY
IBASE_RES_ONE_AT_A_TIME
IBASE_RES_REPLACE
IBASE_RES_CREATE
IBASE_RES_USE_ALL_SPACE
    Options de ibase_restore
IBASE_PRP_PAGE_BUFFERS
IBASE_PRP_SWEEP_INTERVAL
IBASE_PRP_SHUTDOWN_DB
IBASE_PRP_DENY_NEW_TRANSACTIONS
IBASE_PRP_DENY_NEW_ATTACHMENTS
IBASE_PRP_RESERVE_SPACE
IBASE_PRP_RES_USE_FULL
IBASE_PRP_RES
IBASE_PRP_WRITE_MODE
IBASE_PRP_WM_ASYNC
IBASE_PRP_WM_SYNC
IBASE_PRP_ACCESS_MODE
IBASE_PRP_AM_READONLY
IBASE_PRP_AM_READWRITE
IBASE_PRP_SET_SQL_DIALECT
IBASE_PRP_ACTIVATE
IBASE_PRP_DB_ONLINE
IBASE_RPR_CHECK_DB
IBASE_RPR_IGNORE_CHECKSUM
IBASE_RPR_KILL_SHADOWS
IBASE_RPR_MEND_DB
IBASE_RPR_VALIDATE_DB
IBASE_RPR_FULL
IBASE_RPR_SWEEP_DB
    Options de ibase_maintain_db
IBASE_STS_DATA_PAGES
IBASE_STS_DB_LOG
IBASE_STS_HDR_PAGES
IBASE_STS_IDX_PAGES
IBASE_STS_SYS_RELATIONS
    Options de ibase_db_info
IBASE_SVC_SERVER_VERSION
IBASE_SVC_IMPLEMENTATION

```

IBASE_SVC_GET_ENV
IBASE_SVC_GET_ENV_LOCK
IBASE_SVC_GET_ENV_MSG
IBASE_SVC_USER_DBPATH
IBASE_SVC_SVR_DB_INFO
IBASE_SVC_GET_USERS
Options de ibase_server_info

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [ibase_add_user](#)
- [ibase_affected_rows](#)
- [ibase_backup](#)
- [ibase_blob_add](#)
- [ibase_blob_cancel](#)
- [ibase_blob_close](#)
- [ibase_blob_create](#)
- [ibase_blob_echo](#)
- [ibase_blob_get](#)
- [ibase_blob_import](#)
- [ibase_blob_info](#)
- [ibase_blob_open](#)
- [ibase_close](#)
- [ibase_commit_ret](#)
- [ibase_commit](#)
- [ibase_connect](#)
- [ibase_db_info](#)
- [ibase_delete_user](#)
- [ibase_drop_db](#)
- [ibase_errcode](#)
- [ibase_errmsg](#)
- [ibase_execute](#)
- [ibase_fetch_assoc](#)
- [ibase_fetch_object](#)
- [ibase_fetch_row](#)
- [ibase_field_info](#)
- [ibase_free_event_handler](#)
- [ibase_free_query](#)
- [ibase_free_result](#)
- [ibase_gen_id](#)
- [ibase_maintain_db](#)
- [ibase_modify_user](#)
- [ibase_name_result](#)
- [ibase_num_fields](#)
- [ibase_num_params](#)
- [ibase_param_info](#)
- [ibase_pconnect](#)
- [ibase_prepare](#)
- [ibase_query](#)
- [ibase_restore](#)

- [ibase_rollback_ret](#)
- [ibase_rollback](#)
- [ibase_server_info](#)
- [ibase_service_attach](#)
- [ibase_service_detach](#)
- [ibase_set_event_handler](#)
- [ibase_timefmt](#)
- [ibase_trans](#)
- [ibase_wait_event](#)

8.47.8 [ibase_affected_rows\(\)](#) : Retourne le nombre de lignes affectées par la dernière requête iBase

int [ibase_affected_rows](#) (*resource link_identifier*)

[ibase_affected_rows](#) retourne le nombre de lignes qui ont été affectées par la dernière requête (INSERT, UPDATE ou DELETE) qui a été exécuté dans le contexte de transaction spécifié par *link_identifier* . Si *link_identifier* est une ressource de connexion, sa transaction par défaut est utilisée.

Voir aussi [ibase_query](#) et [ibase_execute](#) .

8.47.9 [ibase_backup\(\)](#) : Effectue une sauvegarde de base de données InterBase

mixed [ibase_backup](#) (*resource service_handle* , *string source_db* , *string dest_file* , *int options* , *bool verbose*)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.47.10 [ibase_blob_add\(\)](#) : Ajoute des données dans un BLOB iBase fraîchement créé

void [ibase_blob_add](#) (*resource blob_handle* , *string data*)

[ibase_blob_add](#) ajoute les données *data* dans le BLOB *blob_handle* , créé avec [ibase_blob_create](#) . Retourne NULL en cas de succès ou FALSE si une erreur survient.

Voir aussi [ibase_blob_cancel](#) , [ibase_blob_close](#) , [ibase_blob_create](#) et [ibase_blob_import](#) .

8.47.11 [ibase_blob_cancel\(\)](#) : Annule la création d'un BLOB iBase

bool [ibase_blob_cancel](#) (*resource blob_handle*)

[ibase_blob_cancel](#) annule la création du BLOB *blob_handle* , créé par [ibase_create_blob](#) s'il n'a pas encore été fermé avec [ibase_blob_close](#) . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [ibase_blob_close](#) , [ibase_blob_create](#) et [ibase_blob_import](#) .

8.47.12 [ibase_blob_close\(\)](#) : Ferme un BLOB iBase

mixed [ibase_blob_close](#) (resource blob_handle)

[ibase_blob_close](#) ferme le BLOB `blob_handle` , ouvert en lecture avec [ibase_open_blob](#) ou en écriture avec [ibase_create_blob](#) . Si le BLOB était lu, [ibase_blob_close](#) retourne TRUE en cas de succès, s'il était en cours de modification, [ibase_blob_close](#) retourne une chaîne de caractères contenant l'identifiant du BLOB qui lui a été assigné par la base de données. En cas d'échec, cette fonction retournera FALSE .

Voir aussi [ibase_blob_cancel](#) et [ibase_blob_open](#) .

8.47.13 [ibase_blob_create\(\)](#) : Crée un BLOB iBase pour ajouter des données

resource [ibase_blob_create](#) (resource link_identifieur)

[ibase_blob_create](#) crée un nouveau BLOB à remplir avec des données, sur la connexion InterBase `link_identifieur` . Elle retourne un identifiant de BLOB à utiliser avec [ibase_blob_add](#) ou FALSE en cas d'échec.

Voir aussi [ibase_blob_add](#) , [ibase_blob_cancel](#) , [ibase_blob_close](#) et [ibase_blob_import](#) .

8.47.14 [ibase_blob_echo\(\)](#) : Affiche le contenu d'un BLOB iBase au navigateur

bool [ibase_blob_echo](#) (resource link_identifieur , string blob_id) bool [ibase_blob_echo](#) (string blob_id)

[ibase_blob_echo](#) ouvre le BLOB `blob_id` en lecture et envoie son contenu directement vers la sortie standard (le navigateur dans la plupart des cas). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [ibase_blob_open](#) , [ibase_blob_close](#) et [ibase_blob_get](#) .

8.47.15 [ibase_blob_get\(\)](#) : Lit len octets de données dans un BLOB iBase ouvert

string [ibase_blob_get](#) (resource blob_handle , int len)

[ibase_blob_get](#) retourne au plus `len` octets du BLOB `blob_handle` qui a été ouvert en lecture par [ibase_blob_open](#) . Cette fonction retourne FALSE en cas d'échec.

Exemple avec [ibase_blob_get](#)

```
<?php
    $sql      = "SELECT blob_value FROM matable";
    $result   = ibase_query( $sql );
```

```

$data      = ibase_fetch_object( $result );
$blob_data = ibase_blob_info( $data->BLOB_VALUE );
$blob_hndl = ibase_blob_open( $data->BLOB_VALUE );
echo      ibase_blob_get( $blob_hndl, $blob_data[0] );
?>

```

Cet exemple ne fait pas plus qu'un `ibase_blob_echo($data->BLOB_VALUE)`, mais il montre comment récupérer les informations dans une \$variable pour les manipuler comme vous le voulez.

Note

Il n'est pas possible de lire dans un BLOB ouvert en écriture par `ibase_blob_create`.

Voir aussi `ibase_blob_open`, `ibase_blob_close` et `ibase_blob_echo`.

8.47.16 `ibase_blob_import()` : Crée un BLOB iBase, y copie un fichier et le referme

string `ibase_blob_import` (resource link_identifier , resource file_handle) string
ibase_blob_import (resource file_handle)

`ibase_blob_import` crée un nouveau BLOB sur la connexion iBase `link_identifier`, y copie le fichier `file_handle` en entier, le referme et en retourne l'identifiant assigné. Le pointeur de fichier est une ressource retournée par `fopen`. Cette fonction retourne FALSE en cas d'échec.

Exemple avec `ibase_blob_import`

```

<?php
$dbh = ibase_connect($host, $username, $password);
$filename = '/tmp/bar';

$fd = fopen($filename, 'r');
if ($fd) {

    $blob = ibase_blob_import($dbh, $fd);
    fclose($fd);

    if (!is_string($blob)) {
        // échec de l'importation
    } else {
        $query = "INSERT INTO foo (name, data) VALUES ('$filename', ?)";
        $prepared = ibase_prepare($dbh, $query);
        if (!ibase_execute($prepared, $blob)) {
            // échec de l'insertion de l'enregistrement
        }
    }
} else {
    // impossible d'ouvrir le fichier de données
}
?>

```

Voir aussi `ibase_blob_add`, `ibase_blob_cancel`, `ibase_blob_close` et `ibase_blob_create`.

8.47.17 `ibase_blob_info()` : Retourne la taille d'un BLOB iBase et d'autres informations utiles

array `ibase_blob_info` (resource link_identifier , string blob_id) array **ibase_blob_info** (string blob_id)

ibase_blob_info retourne un tableau contenant des informations à propos du BLOB `blob_id`. Les informations sont la taille du BLOB, le nombre de segments qu'il contient, la taille du segment le plus large, et indiquent s'il s'agit d'un BLOB stream ou segmenté.

8.47.18 **ibase_blob_open()** : Ouvre un BLOB iBase pour récupérer des parties de données

resource **ibase_blob_open** (resource `link_identifieur` , string `blob_id`) resource
ibase_blob_open (string `blob_id`)

ibase_blob_open ouvre le BLOB `blob_id` en lecture. Elle retourne une ressource BLOB à utiliser avec ibase_blob_get ou FALSE en cas d'échec.

Voir aussi ibase_blob_close , ibase_blob_echo et ibase_blob_get .

8.47.19 **ibase_close()** : Ferme une connexion à une base de données Interbase

bool **ibase_close** (resource `connection_id`)

ibase_close ferme une connexion à une base de données Interbase. Cette fonction prend comme argument l'identifiant de connexion `connection_id` retourné par ibase_connect . Si `connection_id` est omis, la dernière connexion iBase est fermée. Les transactions par défaut sont validées et les autres sont annulées. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi ibase_connect et ibase_pconnect .

8.47.20 **ibase_commit_ret()** : Valide une transaction iBase sans la refermer

bool **ibase_commit_ret** (resource `link_or_trans_identifieur`)

Si ibase_commit_ret est appelée sans argument, elle valide la transaction par défaut de la connexion par défaut. Si l'argument `link_identifieur` est un identifiant de connexion, sa transaction par défaut est validée. Si l'argument `link_identifieur` est un identifiant de transaction, celle-ci sera validée. Le contexte de transaction sera retenu et, donc, les requêtes exécutées dans cette transaction ne seront pas invalidées. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.47.21 **ibase_commit()** : Valide une transaction iBase

bool **ibase_commit** (resource `link_or_trans_identifieur`)

ibase_commit valide la transaction `link_or_trans_identifieur` , qui a été préparée avec ibase_trans . Si `link_or_trans_identifieur` est une ressource de connexion, la transaction correspondante sera validée. Si l'argument est une ressource de transaction, cette transaction sera validée. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.47.22 `ibase_connect()` : Ouvre une connexion à une base de données InterBase

resource **ibase_connect** (*string database* , *string username* , *string password* , *string charset* , *int buffers* , *int dialect* , *string role* , *int sync*)

`ibase_connect` établit une connexion avec un serveur InterBase. `database` doit être un chemin valide jusqu'à un fichier de base de données sur le serveur sur lequel il réside. Si le serveur est distant, il faut le préfixer avec un nom d'hôte 'hostname:' (TCP/IP), '//hostname/' (NetBEUI) ou 'hostname@' (IPX/SPX), en fonction du protocole de communication utilisé. `username` et `password` peuvent être spécifiés dans les directives de configuration de PHP `ibase.default_user` et `ibase.default_password`. `charset` est le jeu de caractère par défaut de la base. `buffers` est le nombre de buffer de base à allouer pour le cache serveur. S'il est passé à 0 ou omis, le serveur le choisira de lui-même. `dialect` sélectionne le dialecte SQL pour les requêtes exécutées avec cette connexion et, par défaut, il utilise le meilleur dialecte disponible.

Si un deuxième appel est fait avec `ibase_connect` , en passant les mêmes arguments, une nouvelle connexion ne sera pas ouverte, mais la connexion déjà ouverte sera retournée. La connexion sera fermée dès que le script se termine, à moins qu'elle ne soit fermée explicitement avec `ibase_close` , durant le script.

Exemple `ibase_connect`

```
<?php
    $dbh = ibase_connect($host, $username, $password);
    $stmt = 'SELECT * FROM tblname';
    $sth = ibase_query($dbh, $stmt);
    while ($row = ibase_fetch_object($sth)) {
        echo $row->email . "\n";
    }
    ibase_close($dbh);
?>
```

Note

`buffers` a été ajouté en PHP 4.0.0.

Note

`dialect` a été ajouté en PHP 4.0.0. Il n'est opérationnel qu'avec les versions InterBase 6 et plus récentes.

Note

`role` a été ajouté en PHP 4.0.0. Il n'est opérationnel qu'avec les versions InterBase 5 et plus récentes.

Note

Si vous avez quelques erreurs comme "arithmetic exception, numeric overflow, or string truncation. Cannot transliterate character between character sets" (cela se produit lorsque vous tentez d'utiliser quelques caractères accentués) lorsque vous utilisez `ibase_connect` et après `ibase_query` , vous devez positionner un `charset` correct (i.e. ISO8859_1 ou votre `charset` courant).

Voir aussi `ibase_pconnect` et `ibase_close` .

8.47.23 `ibase_db_info()` : Demande des statistiques sur une base de données Interbase

string **ibase_db_info** (resource service_handle , string db , int action , int argument)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.47.24 `ibase_delete_user()` : Efface un utilisateur d'une base de données de sécurité (uniquement pour IB6 ou plus récent)

bool `ibase_delete_user` (resource `service_handle` , string `user_name`)

PHP 4 utilise les paramètres `server` , `dba_user_name` et `dba_user_password` au lieu de `service_handle` .

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Voir aussi [ibase_add_user](#) et [ibase_modify_user](#) .

8.47.25 `ibase_drop_db()` : Supprime une base de données iBase

bool `ibase_drop_db` (*resource connection*)

`ibase_drop_db` supprime une base de données qui a été ouverte par [ibase_connect](#) ou [ibase_pconnect](#) . La base de données est refermée et supprimée du serveur. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [ibase_connect](#) et [ibase_pconnect](#) .

8.47.26 `ibase_errcode()` : Retourne le code d'erreur iBase

int `ibase_errcode` (void)

`ibase_errcode` retourne le code erreur résultant de l'appel le plus récent à une fonction InterBase. `ibase_errcode` retourne FALSE si aucune erreur n'est survenue.

Voir aussi [ibase_errmsg](#) .

8.47.27 `ibase_errmsg()` : Retourne un message d'erreur

string `ibase_errmsg` (void)

`ibase_errmsg` retourne une chaîne contenant les messages d'erreurs. Retourne FALSE si aucune erreur n'est survenu.

8.47.28 `ibase_execute()` : Exécute une requête iBase préparée

resource `ibase_execute` (resource `query` , *mixed bind_arg* , *mixed ...*)

`ibase_execute` exécute une requête préparée (et éventuellement liée) par [ibase_prepare](#) . `ibase_execute` est beaucoup plus efficace que [ibase_query](#) , si vous effectuez plusieurs fois la même requête en ne changeant que quelques paramètres.

Exemple avec `ibase_execute`

```

<?php

$dbh = ibase_connect($host, $username, $password);

$update = array(
    1 => 'Eric',
    5 => 'Filip',
    7 => 'Larry'
);

$query = ibase_prepare($dbh, "UPDATE FOO SET BAR = ? WHERE BAZ = ?");

foreach ($update as $baz => $bar) {
    ibase_execute($query, $bar, $baz);
}

?>

```

Note

Depuis la version 5.0.0 de PHP, `ibase_execute` retourne le nombre d'enregistrements affectés par la requête (si supérieur à 0). Pour une requête qui réussie mais qui ne renvoie aucun enregistrement (e.g un UPDATE sur un enregistrement inexistant), `ibase_execute` retournera TRUE.

Voir aussi `ibase_query`.

8.47.29 `ibase_fetch_assoc()` : Récupère une ligne du résultat d'une requête dans un tableau associatif

array `ibase_fetch_assoc` (resource result , int fetch_flag)

`ibase_fetch_assoc` retourne un tableau associatif qui correspond à la ligne récupérée. Les appels suivants renvoient la ligne qui suit dans le jeu de résultats, ou FALSE s'il n'y a plus de lignes.

`ibase_fetch_assoc` récupère une ligne de données à partir de result . Si deux ou plus colonnes ont le même nom de champ, la dernière colonne prendra précedence. Pour accéder aux autres colonnes du même nom, vous devez soit le faire avec leurs indices numériques en utilisant `ibase_fetch_row` ou en utilisant des alias dans votre requête.

fetch_flag est une combinaison des constantes IBASE_TEXT et IBASE_UNIXTIME. Passer IBASE_TEXT fait retourner le contenu du BLOB au lieu de l'ID du BLOB. Passer IBASE_UNIXTIME fait retourner les valeurs date/time sous forme de timestamps UNIX au lieu de chaînes formatées.

Voir aussi `ibase_query` , `ibase_fetch_row` , et `ibase_fetch_object` .

8.47.30 `ibase_fetch_object()` : Lit une ligne dans une base Interbase dans un objet

object `ibase_fetch_object` (resource result_id , int fetch_flag)

`ibase_fetch_object` lit une ligne dans une base Interbase et la place dans un pseudo objet. `ibase_fetch_object` prend comme argument l'identifiant de résultat result_id obtenu de `ibase_query` ou `ibase_execute` .

Exemple avec `ibase_fetch_object`

```
<?php
$dbh = ibase_connect ($host, $username, $password);
$stmt = 'SELECT * FROM tblname';
$stmt = ibase_query ($dbh, $stmt);
while ($row = ibase_fetch_object ($sth)) {
    echo $row->email . "\n";
}
ibase_close ($dbh);
?>
```

L'appel à `ibase_fetch_object` renvoie la ligne suivante dans le jeu d'enregistrements ou FALSE s'il n'y a plus de lignes disponibles.

`fetch_flag` est une combinaison des constantes `IBASE_TEXT` et `IBASE_UNIXTIME`. Passer `IBASE_TEXT` fait retourner le contenu du BLOB au lieu de l'ID du BLOB. Passer `IBASE_UNIXTIME` fait retourner les valeurs date/time sous forme de timestamps UNIX au lieu de chaînes formatées.

Voir aussi `ibase_fetch_row` et `ibase_fetch_assoc` .

8.47.31 `ibase_fetch_row()` : Lit une ligne dans une base Interbase

array `ibase_fetch_row` (resource result_identifiant , int fetch_flag)

`ibase_fetch_row` retourne la prochaine ligne spécifiée dans le résultat obtenu de `ibase_query` , ou FALSE s'il n'y a plus de lignes.

`ibase_fetch_row` lit une ligne de données dans le résultat, sous la forme d'un tableau numérique. Chaque colonne est stockée avec un offset différent, commençant à 0.

Les appels suivants à `ibase_fetch_row` vont retourner la prochaine ligne dans le résultat, ou bien FALSE s'il n'y a plus de ligne.

`fetch_flag` est une combinaison des constantes `IBASE_TEXT` et `IBASE_UNIXTIME`. Passer `IBASE_TEXT` fait retourner le contenu du BLOB au lieu de l'ID du BLOB. Passer `IBASE_UNIXTIME` fait retourner les valeurs date/time sous forme de timestamps UNIX au lieu de chaînes formatées.

Voir aussi `ibase_fetch_assoc` et `ibase_fetch_row` .

8.47.32 `ibase_field_info()` : Lit les informations sur un champ iBase

array `ibase_field_info` (resource result , int field_number)

`ibase_field_info` retourne un tableau contenant les informations sur le champ numéro `field_number` après une requête de SELECT. Le tableau contient les index name (nom), alias , relation , length (taille), type .

Exemple avec `ibase_field_info`

```
<?php
$rs = ibase_query("Select * from matable");
$coln = ibase_num_fields($rs);
for ($i=0 ; $i < $coln ; $i++) {
    $col_info = ibase_field_info($rs, $i);
    echo "Nom : " . $col_info['name'] . "\n";
}
```

```

    echo "Alias : " . $col_info['alias'] . "\n";
    echo "Relation : " . $col_info['relation'] . "\n";
    echo "Taille : " . $col_info['length'] . "\n";
    echo "Type : " . $col_info['type'] . "\n";
}
?>

```

Voir aussi [ibase_num_fields](#) .

8.47.33 `ibase_free_event_handler()` : Libère un gestionnaire d'événements iBase

bool `ibase_free_event_handler` (resource event)

`ibase_free_event_handler` annule le gestionnaire d'événements enregistré, spécifié par event . La fonction de callback ne sera plus exécutée pour les événements qu'elle devait prendre en charge. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [ibase_set_event_handler](#) .

8.47.34 `ibase_free_query()` : Libère la mémoire réservée par une requête préparée

bool `ibase_free_query` (resource query)

`ibase_free_query` libère la mémoire réservée par une requête préparée par `ibase_prepare` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.47.35 `ibase_free_result()` : Libère un résultat iBase

bool `ibase_free_result` (resource result_identifieur)

`ibase_free_result` libère les ressources liées au résultat result_identifieur . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.47.36 `ibase_gen_id()` : Incrémente le générateur donné et retourne sa nouvelle valeur

mixed `ibase_gen_id` (string generator , int increment , resource link_identifieur)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Retourne une nouvelle valeur générée sous la forme d'un entier ou une chaîne si la valeur est trop grosse.

8.47.37 `ibase_maintain_db()` : Exécute une commande de maintenance sur une base de données Interbase

bool `ibase_maintain_db` (resource `service_handle` , string `db` , int `action` , int *argument*)

Attention
 Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.47.38 `ibase_modify_user()` : Modifie un utilisateur dans une base de données de sécurité (uniquement pour InterBase6 ou plus récent)

bool `ibase_modify_user` (resource `service_handle` , string `user_name` , string `password` , string *first_name* , string *middle_name* , string *last_name*)

PHP 4 utilise les paramètres `server` , `dba_user_name` et `dba_user_password` au lieu de `service_handle` .

Attention
 Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Voir aussi [ibase_add_user](#) et [ibase_delete_user](#) .

8.47.39 `ibase_name_result()` : Assigne un nom à un jeu de résultats iBase

bool `ibase_name_result` (resource `result` , string `name`)

`ibase_name_result` assigne un nom à un jeu de résultats. Ce nom peut être utilisé plus tard dans les requêtes de type UPDATE|DELETE ... WHERE CURRENT OF `name` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec `ibase_name_result`

```
<?php
$result = ibase_query ("SELECT field1,field2 FROM table FOR UPDATE" );
ibase_name_result ($result,"my_cursor");

$updateqry = ibase_prepare ("UPDATE table SET field2 = ? WHERE CURRENT OF my_cursor");

for ($i = 0; ibase_fetch_row ($result); ++$i) {
    ibase_execute ($updateqry,$i);
}
?>
```

Voir aussi [ibase_prepare](#) et [ibase_execute](#) .

8.47.40 `ibase_num_fields()` : Retourne le nombre de colonnes dans un résultat iBase

int `ibase_num_fields` (resource `result_id`)

`ibase_num_fields` retourne le nombre de colonnes dans le résultat iBase `result_id` .

Exemple avec `ibase_num_fields`

```

<?php
$rs = ibase_query("SELECT * FROM matable");
$coln = ibase_num_fields($rs);
for ($i = 0; $i < $coln; $i++) {
    $col_info = ibase_field_info($rs, $i);
    echo "Nom : " . $col_info['name'] . "\n";
    echo "Alias : " . $col_info['alias'] . "\n";
    echo "Relation : " . $col_info['relation'] . "\n";
    echo "Taille : " . $col_info['length'] . "\n";
    echo "Type : " . $col_info['type'] . "\n";
}
?>

```

Voir aussi [ibase_field_info](#) .

8.47.41 `ibase_num_params()` : Retourne le nombre de paramètres dans une requête préparée iBase

int `ibase_num_params` (resource query)

`ibase_num_params` retourne le nombre de paramètres dans la requête spécifiée par query . C'est le nombre d'arguments de substitution qui doivent être présents lors de l'appel de [ibase_execute](#) .

Voir aussi [ibase_prepare](#) et [ibase_param_info](#) .

8.47.42 `ibase_param_info()` : Retourne des informations à propos d'un paramètre dans une requête préparée iBase

array `ibase_param_info` (resource query , int param_number)

`ibase_param_info` retourne un tableau avec des informations à propos d'un paramètre après qu'une requête n'ait été préparée. Le tableau contient les entrées name , alias , relation , length , type (nom, alias, relation, taille, type).

Voir aussi [ibase_field_info](#) et [ibase_num_params](#) .

8.47.43 `ibase_pconnect()` : Ouvre une connexion persistante à une base de données InterBase

resource `ibase_pconnect` (string database , string username , string password , string charset , int buffers , int dialect , string role , int sync)

`ibase_pconnect` se comporte similairement à [ibase_connect](#) , avec deux différences majeures : la première est que, lors de la connexion, la fonction va essayer de trouver une connexion (persistante) déjà ouverte. Si elle la trouve, cette dernière sera retournée, plutôt qu'une nouvelle connexion. Sinon, une nouvelle connexion sera ouverte. La deuxième est que la connexion ne sera pas fermée à la fin du script, mais restera ouverte pour utilisation ultérieure. ([ibase_close](#) ne fermera pas une connexion ouverte avec [ibase_pconnect](#)). Ce type de lien est alors dit 'persistant'.

Note

buffers a été ajouté en PHP 4.0.0.

Note

dialect a été ajouté en PHP 4.0.0. Il n'est opérationnel qu'avec les versions InterBase 6 et plus récentes.

Note

role a été ajouté en PHP 4.0.0. Il n'est opérationnel qu'avec les versions InterBase 5 et plus récentes.

Voir aussi [ibase_close](#) et [ibase_connect](#) pour plus de détails sur les arguments de cette fonction.

8.47.44 [ibase_prepare\(\)](#) : Prépare une requête iBase pour lier les paramètres et l'exécuter ultérieurement

resource [ibase_prepare](#) (string query) resource [ibase_prepare](#) (resource link_identifieur , string query) resource [ibase_prepare](#) (resource link_identifieur , string trans , string query)

[ibase_prepare](#) prépare une requête pour l'exécuter

8.47.45 [ibase_query\(\)](#) : Exécute une requête sur une base iBase

resource [ibase_query](#) (resource link_identifieur , string query , int bind_args)

[ibase_query](#) exécute la requête query sur une base Interbase, et retourne un identifiant de résultat, à utiliser avec [ibase_fetch_row](#) , [ibase_free_result](#) et/ou [ibase_free_query](#) . [ibase_query](#) retourne FALSE si la requête échoue.

Exemple avec [ibase_query](#)

```
<?php
    $host = 'localhost:/path/to/your.gdb';

    $dbh = ibase_connect($host, $username, $password);
    $stmt = 'SELECT * FROM tblname';

    $sth = ibase_query($dbh, $stmt) or die(ibase_errmsg());
?>
```

Note

Dans les version 5.0.0 de PHP et suivantes, [ibase_query](#) retourne le nombre d'enregistrements affectés par les requêtes INSERT , UPDATE et DELETE . Dans un souci de compatibilité ascendante, [ibase_query](#) retournera TRUE si la requête réussit mais ne renvoie aucun enregistrement.

Note

Si vous recevez une erreur du type "arithmetic exception, numeric overflow, or string truncation. Cannot transliterate character between character sets" (ceci survient lorsque vous essayez d'utiliser des caractères accentués) avec la fonction [ibase_query](#) , c'est que vous devez choisir un jeu de caractères (i.e. ISO8859_1 ou votre jeu courant).

Voir aussi [ibase_errmsg](#) , [ibase_fetch_row](#) , [ibase_fetch_object](#) et [ibase_free_result](#) .

8.47.46 `ibase_restore()` : Restaure une sauvegarde de base de données Interbase

mixed **ibase_restore** (resource service_handle , string source_file , string dest_db , int options , bool verbose)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.47.47 `ibase_rollback_ret()` : Annule une transaction sans la refermer

bool **ibase_rollback_ret** (resource link_or_trans_identifieur)

Si `ibase_rollback_ret` est appelée sans argument, elle annule la transaction par défaut du lien par défaut. Si l'argument `link_identifieur` est un identifiant de connexion, la transaction par défaut de cette connexion sera annulée. Si l'argument `link_identifieur` est un identifiant de transaction, celle-ci sera annulée. Le contexte de transaction sera retenu et, donc, les requêtes exécutées dans cette transaction ne seront pas invalidées. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.47.48 `ibase_rollback()` : Annule une transaction iBase

bool **ibase_rollback** (resource link_or_trans_identifieur)

Lorsque `ibase_rollback` est appelée sans argument, elle annule la transaction par défaut du lien par défaut. Si l'argument `link_or_trans_identifieur` est un identifiant de connexion, la transaction par défaut de cette connexion sera annulée. Si l'argument `link_identifieur` est un identifiant de transaction, celle-ci sera annulée. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.47.49 `ibase_server_info()` : Demande des informations sur le serveur Interbase

string **ibase_server_info** (resource service_handle , int action)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.47.50 `ibase_service_attach()` : Connexion au service de gestion Interbase

resource **ibase_service_attach** (string host , string dba_username , string dba_password)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.47.51 `ibase_service_detach()` : Déconnexion du service de gestion Interbase

bool `ibase_service_detach` (resource service_handle)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.47.52 `ibase_set_event_handler()` : Enregistre une fonction de callback sur un événement iBase

resource `ibase_set_event_handler` (callback event_handler , string event_name1 , string event_name2 , string ...) resource `ibase_set_event_handler` (resource connection , callback event_handler , string event_name1 , string event_name2 , string ...)

`ibase_set_event_handler` enregistre la fonction utilisateur PHP `event_handler` en tant que gestionnaire d'événements pour les événements spécifiés par `event_name1` , `event_name2` ... La fonction `event_handler` callback est appelée avec le nom de l'événement et la ressource de connexion en tant qu'arguments dès qu'un événement spécifié est posté dans la base de données. La fonction callback doit retourner FALSE si le gestionnaire doit être annulé. Toute autre valeur de retour est ignorée. Cette fonction accepte jusqu'à 15 arguments d'événement.

Exemple avec `ibase_set_event_handler`

```
<?php
function event_handler($event_name, $link) {
    if ($event_name=="NEW ORDER") {
        // Traitement du nouvel ordre
        ibase_query($link, "UPDATE orders SET status='handled'");
    } else if ($event_name=="DB_SHUTDOWN") {
        // Libération du gestionnaire
        return false;
    }
}
ibase_set_event_handler($link,"event_handler","NEW_ORDER","DB_SHUTDOWN");
?>
```

La valeur retournée est une ressource d'événement. Elle peut être utilisée pour libérer le gestionnaire d'événements en utilisant `ibase_free_event_handler` .

Voir aussi `ibase_free_event_handler` et `ibase_wait_event` .

8.47.53 `ibase_timefmt()` : Fixe le format de date pour les prochaines requêtes

int `ibase_timefmt` (string format , int columntype)

`ibase_timefmt` fixe le format des colonnes de type date, heure et timestamp retournées par les requêtes. En interne, les colonnes sont formatées par la fonction C `strftime()` : reportez-vous à sa documentation pour connaître la structure de la chaîne de format. `columntype` est une des constantes suivantes : `IBASE_TIMESTAMP`, `IBASE_DATE` ou `IBASE_TIME`. Si elle est omise, la valeur par défaut est `IBASE_TIMESTAMP`, pour compatibilité ascendante.

Exemple avec `ibase_timefmt`

```
<?php
/* Les colonnes TIME de InterBase 6 seront retournées avec
 * la forme '05 heures 37 minutes'. */
ibase_timefmt("%H heures %M minutes", IBASE_TIME);
?>
```

Vous pouvez aussi modifier les formats par défaut avec les directives PHP `ibase.timestampformat`, `ibase.dateformat` et `ibase.timeformat`.

Note

Cette fonction a été supprimée de PHP 5, utilisez `ini_set` à la place.

8.47.54 `ibase_trans()` : Prépare une transaction iBase

resource **ibase_trans** (*int trans_args* , *resource link_identifieur*)

`ibase_trans` prépare une transaction sur la connexion InterBase `link_identifieur` .

`trans_args` peut être une combinaison des constantes suivantes : `IBASE_READ` , `IBASE_WRITE` , `IBASE_COMMITTED` , `IBASE_CONSISTENCY` , `IBASE_CONCURRENCY` , `IBASE_REC_VERSION` , `IBASE_REC_NO_VERSION` , `IBASE_WAIT` et `IBASE_NOWAIT` .

Note

Le comportement de cette fonction a changé depuis PHP 5.0.0. Le premier appel à `ibase_trans` retournera la transaction par défaut pour la connexion courante. Toutes les transactions commencées par `ibase_trans` seront annulées à la fin de l'exécution du script si elles n'ont pas été validées ou annulées par respectivement les fonctions `ibase_commit` ou `ibase_rollback` .

Note

Dans les versions 5.0.0 de PHP et suivantes, `ibase_trans` accepte plusieurs paramètres `trans_args` et `link_identifieur` . Ceci permet d'effectuer des transactions sur plusieurs connexions à des bases de données différentes, qui seront validées en utilisant l'algorithme 2-phase . Cela signifie que vous pouvez mettre à jour plusieurs bases de données. Cela ne veut PAS dire que vous pouvez utiliser plusieurs bases de données dans une même requête !

Si vous utilisez les transactions sur plusieurs base de données, vous devez spécifier `link_id` et `transaction_id` dans les fonctions `ibase_query` et `ibase_prepare` .

8.47.55 `ibase_wait_event()` : Attend un événement iBase

string **ibase_wait_event** (string *event_name1* , string *event_name2* , string ...) string
ibase_wait_event (resource *connection* , string *event_name1* , string *event_name2* , string ...)

`ibase_wait_event` suspend l'exécution du script jusqu'à ce que l'un des événements spécifiés par `event_name1` , `event_name2` , ... soit posté par la base de données. Le nom de l'événement qui a été posté est alors retourné. Cette fonction accepte jusqu'à 15 arguments événements.

Voir aussi `ibase_set_event_handler` et `ibase_free_event_handler` .

8.48 ICAP

8.48.1 Introduction

Note

L'extension icap sera supprimée dans un futur proche. Le module icap et la bibliothèque ne sont plus supportés. Si vous voulez utiliser ces fonctions de calendrier avec PHP, utilisez plutôt [mcal](#) .

8.48.2 Pré-requis

La bibliothèque icap doit être installée. Elle n'est plus supportée, ni disponible.

8.48.3 Installation

Pour faire fonctionner cette extension, vous devez compiler PHP avec l'option `--with-icap` .

8.48.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.48.5 Types de ressources

Cette extension ne définit aucune ressource.

8.48.6 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [icap_close](#)
- [icap_create_calendar](#)
- [icap_delete_calendar](#)
- [icap_delete_event](#)
- [icap_fetch_event](#)
- [icap_list_alarms](#)
- [icap_list_events](#)
- [icap_open](#)
- [icap_rename_calendar](#)
- [icap_reopen](#)
- [icap_snooze](#)
- [icap_store_event](#)

8.48.8 `icap_create_calendar()` : Crée un nouveau calendrier

string `icap_create_calendar` (int `stream_id` , string `calendar`)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.48.9 `icap_delete_calendar()` : Supprime un calendrier

string `icap_delete_calendar` (int `stream_id` , string `calendar`)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.48.10 `icap_delete_event()` : Efface un événement dans un agenda ICAP

string `icap_delete_event` (int `stream_id` , int `uid`)

`icap_delete_event` efface l'événement d'identifiant `uid` .

`icap_delete_event` retourne TRUE .

8.48.11 `icap_fetch_event()` : Recherche un événement dans le calendrier

int `icap_fetch_event` (int `stream_id` , int `event_id` , int `options`)

`icap_fetch_event` recherche un événement dans le calendrier spécifié par `id` .

`icap_fetch_event` retourne un objet événement dont les attributs sont :

- int `id` - ID de l'événement.
- int `public` - TRUE si l'événement est public, FALSE s'il est privé.
- string `category` - Catégorie de l'événement.
- string `title` - Titre de l'événement.
- string `description` - Description de l'événement.
- int `alarm` - Nombre de minutes avant d'envoyer une alerte pour cet événement.
- object `start` - Objet contenant une date et une heure.
- object `end` - Objet contenant une date et une heure.

Tous les objets de date et heure sont construits comme suit :

- int `year` - année
- int `month` - mois
- int `mday` - jour du mois
- int `hour` - heure
- int `min` - minutes
- int `sec` - secondes

8.48.12 `icap_list_alarms()` : Retourne une liste d'événements qui ont une alarme prévue à une date

int `icap_list_alarms` (int `stream_id` , array `date` , array `time`)

`icap_list_alarms` retourne un tableau d'identifiants, qui ont une alarme de prévue à la date `alarm_date` .

`icap_list_alarms` prend une date, et retourne un tableau d'identifiants.

Tous les objets de date et heure sont construits comme suit :

- int `year` - année
- int `month` - mois
- int `mday` - jour du mois
- int `hour` - heure
- int `min` - minutes
- int `sec` - secondes

8.48.13 `icap_list_events()` : Retourne une liste d'événements entre deux dates

array `icap_list_events` (int `stream_id` , int `begin_date` , int `end_date`)

`icap_list_events` retourne un tableau d'identifiants d'événements compris entre deux dates.

`icap_list_events` prend une date de début et une date de fin. Un tableau d'identifiants est retourné.

Tous les objets de date et heure sont construits comme suit :

- int `year` : année
- int `month` : mois
- int `mday` : jour du mois
- int `hour` : heure
- int `min` : minutes
- int `sec` : secondes

8.48.14 `icap_open()` : Ouvre une connexion ICAP

resource `icap_open` (string `calendar` , string `username` , string `password` , string `options`)

`icap_open` retourne un flux ICAP en cas de succès, FALSE en cas d'erreur.

`icap_open` ouvre une connexion ICAP au serveur de calendrier `calendar` . Si le paramètre `options` est spécifié, passe `options` à la boîte aux lettres (????).

8.48.15 `icap_rename_calendar()` : Renomme un calendrier

string `icap_rename_calendar` (int `stream_id` , string `old_name` , string `new_name`)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.48.16 `icap_reopen()` : Réouvre un flux ICAP pour un nouveau calendrier

`int icap_reopen (int stream_id , string calendar , int options)`

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.48.17 `icap_snooze()` : Eteind l'alarme d'un événement

`string icap_snooze (int stream_id , int uid)`

`icap_snooze` éteind l'alarme de l'événement identifié par l'UID `uid` .

`icap_snooze` retourne TRUE .

8.48.18 `icap_store_event()` : Enregistre un événement dans un agenda ICAP

`string icap_store_event (int stream_id , object event)`

`icap_store_event` enregistre un événement dans un calendrier ICAP.

Un événement est un objet avec les attributs suivants :

- `int public` - 1 si l'événement est public, 0 s'il est privé.
- `string category` - Catégorie de l'événement.
- `string title` - Titre de l'événement.
- `string description` - Description de l'événement.
- `int alarm` - Nombre de minutes avant d'envoyer une alerte pour cet événement.
- `object start` - Objet contenant une date et une heure.
- `object end` - Objet contenant une date et une heure.

Tous les objets de date et heure sont construits comme suit :

- `int year` : année
- `int month` : mois
- `int mday` : jour du mois
- `int hour` : heure
- `int min` : minutes
- `int sec` : secondes

`icap_store_event` retourne TRUE en cas de succès, et FALSE en cas d'erreur.

8.49 Iconv

8.49.1 Introduction

Ce module est une interface vers la bibliothèque iconv. L'extension iconv convertit des fichiers entre divers jeux de caractères. Les jeux supportés dépendent de l'implémentation de iconv() sur votre système. Notez que cette fonction ne fonctionne pas toujours bien sur tous les systèmes. Dans ce cas, ce serait une bonne idée d'installer la bibliothèque [GNU libiconv](#) .

Depuis PHP 5.0.0, cette extension dispose de beaucoup de fonctions utiles qui peuvent vous aider à écrire des scripts multilingues. Regardez les sections suivantes pour voir des nouvelles fonctionnalités.

8.49.2 Pré-requis

Vous n'avez besoin de rien de spécial si votre système est conforme au standard POSIX car la bibliothèque standard C fournit iconv. Dans le cas contraire, vous devez installer la bibliothèque [libiconv](#) sur votre système.

8.49.3 Installation

Pour pouvoir l'utiliser, vous devez compiler PHP avec l'option `--with-iconv[=DIR]` .

Note

Note aux utilisateurs Win32

Afin d'activer ce module dans l'environnement Windows®, vous devez copier la bibliothèque **iconv.dll** ou **iconv-1.3.dll** (pour les versions antérieures à 4.2.1) qui est fournie avec le package PHP/Win32 dans un dossier spécifié par la variable d'environnement PATH ou dans un des dossiers systèmes de votre installation de Windows®.

Ce module fait parti de PHP depuis PHP 4, donc les bibliothèques **iconv.dll** et **php_iconv.dll** ne sont plus nécessaires.

8.49.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

Nom	Par défaut	Modifiable	Historique
iconv.input_encoding	"ISO-8859-1"	PHP_INI_ALL	Disponible depuis PHP 4.0.5.
iconv.output_encoding	"ISO-8859-1"	PHP_INI_ALL	Disponible depuis PHP 4.0.5.
iconv.internal_encoding	"ISO-8859-1"	PHP_INI_ALL	Disponible depuis PHP 4.0.5.

Pour plus de détails sur les constantes PHP INI *, reportez-vous à [Directives du php.ini](#) .

Attention

Quelques systèmes (comme IBM AIX) utilisent "ISO8859-1" au lieu de "ISO-8859-1", cette valeur doit donc être utilisée dans les options de configuration ainsi que dans les paramètres des fonctions.

Note

L'option de configuration <code>iconv.input_encoding</code> n'est actuellement pas utilisée.
--

8.49.5 Types de ressources

Cette extension ne définit aucune ressource.

8.49.6 Constantes pré-définies

Depuis PHP 4.3.0, il est possible d'identifier durant l'exécution, la version de la bibliothèque `iconv` que vous utilisez.

Constante	Type	Description
<code>ICONV_IMPL</code>	string	Le nom de la bibliothèque
<code>ICONV_VERSION</code>	string	La version de la bibliothèque
Note		
La programmation de scripts dépendant de versions spécifiques, avec ces constantes, est fortement déconseillée.		

Depuis PHP 5.0.0, les constantes suivantes sont également disponibles :

Constante	Type	Description
<code>ICONV_MIME_DECODE_STRICT</code>	entier	Un masque utilisé par <code>iconv_mime_decode</code>
<code>ICONV_MIME_DECODE_CONTINUE_ON_ERROR</code>	entier	Un masque utilisé pour <code>iconv_mime_decode</code>

8.49.7 Voir aussi

Voir aussi les fonctions [GNU Recode](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Voir aussi](#)
- [iconv_get_encoding](#)
- [iconv_mime_decode_headers](#)
- [iconv_mime_decode](#)
- [iconv_mime_encode](#)
- [iconv_set_encoding](#)
- [iconv_strlen](#)
- [iconv_strpos](#)
- [iconv_strrpos](#)
- [iconv_substr](#)
- [iconv](#)
- [ob_iconv_handler](#)

8.49.9 iconv_mime_decode_headers() : Décode des en-têtes MIME multiples

array **iconv_mime_decode_headers** (string *encoded_headers* , int *mode* , string *charset*)

iconv_mime_decode_headers retourne un tableau associatif qui contient les en-têtes MIME spécifiés par le paramètre *encoded_headers* , ou bien FALSE si une erreur survient durant le décodage.

Chaque clé du tableau retourné contient un nom d'en-tête distinct, et sa valeur correspondante. Si plusieurs champs ont le même nom, iconv_mime_decode_headers fera de ce champ un tableau indexé, avec les valeurs dans leur ordre d'apparence.

mode détermine le comportement de la fonction, si iconv_mime_decode_headers rencontre un en-tête MIME malformé. Vous pouvez spécifier une combinaison des options suivantes :

Valeur	Constante	Description
1	ICONV_MIME_DECODE_STRICT	Si utilisés, les en-têtes sont décodés en respectant scrupuleusement le standard de la RFC2047 . Cette option est désactivée par défaut, car il y a de nombreux clients mails qui ne suivent pas ces spécifications et qui ne produisent pas d'en-têtes MIME corrects.
2	ICONV_MIME_DECODE_CONTINUE_ON_ERROR	Si cette option est activée, <u>iconv_mime_decode_headers</u> tente d'ignorer les erreurs de syntaxe et continue de traiter l'en-tête donné.

Le paramètre optionnel *charset* spécifie le jeu de caractères utilisé pour représenter le résultat. S'il est omis, le jeu définit dans le fichier `php.ini` `iconv.internal_charset` est utilisé.

Exemple avec <u>iconv_mime_decode_headers</u>
<pre><?php \$headers_string = <<<EOF Subject: =?UTF-8?B?UHLdVgZ1bmcgUHLdVgZ1bmc=?= To: example@example.com Date: Thu, 1 Jan 1970 00:00:00 +0000 Message-Id: <example@example.com> Received: from localhost (localhost [127.0.0.1]) by localhost with SMTP id example for <example@example.com> Thu, 1 Jan 1970 00:00:00 +0000 (UTC) (envelope-from example-return-0000-example@example.com@example.com) Received: (qmail 0 invoked by uid 65534); 1 Thu 2003 00:00:00 +0000 EOF; \$headers = iconv_mime_decode_headers(\$headers_string, 0, "ISO-8859-1"); print_r(\$headers); ?></pre> <p>L'exemple ci-dessus va afficher :</p> <p>Array</p>

```
(
[Subject] => Prüfung Prüfung
[To] => example@example.com
[Date] => Thu, 1 Jan 1970 00:00:00 +0000
[Message-Id] => <example@example.com>
[Received] => Array
(
[0] => from localhost (localhost [127.0.0.1]) by localhost with SMTP id example for
<example@example.com>; Thu, 1 Jan 1970 00:00:00 +0000 (UTC) (envelope-from
example-return-0000-example@example.com@example.com)
[1] => (qmail 0 invoked by uid 65534); 1 Thu 2003 00:00:00 +0000
)
)
)
```

Voir aussi [iconv_mime_decode](#) , [mb_decode_mimeheader](#) , [imap_mime_header_decode](#) , [imap_base64](#) et [imap_qprint](#) .

8.49.10 iconv_mime_decode() : Décode un champ d'en-tête MIME

string **iconv_mime_decode** (string encoded_header , int mode , string charset)

[iconv_mime_decode](#) retourne un champ d'en-tête décodé ou FALSE si une erreur survient.

mode détermine une alternative dans le cas où [iconv_mime_decode](#) rencontre un champ d'en-tête MIME mal formé. Pour pouvoir spécifier n'importe lesquels des masques suivantes :

Valeur	Constante	Description
1	ICONV_MIME_DECODE_STRICT	Si défini, l'en-tête correspondante sera décodé en suivant strictement le standard RFC2047 . Cette option est désactivé par défaut car il existe beaucoup de mauvais mail user agents qui ne suivent pas ce standard et donc, produisent de mauvaises en-têtes MIME .
2	ICONV_MIME_DECODE_CONTINUE_ON_ERROR	Si défini, iconv_mime_decode essaie de continuer à décoder l'en-tête passée même si des erreurs apparaissent.

Le paramètre par défaut charset spécifie le jeux de caractères a utilisé pour représenter le résultat. S'il est omis, [iconv.internal_charset](#) sera utilisé.

Exemple avec [iconv_mime_decode](#)

```
<?php
// Ceci affichera : "Subject: Prüfung Prüfung"
echo iconv_mime_decode("Subject: =?UTF-8?B?UHLdVgZ1bmcgUHLdVgZ1bmc=?=",
0, "ISO-8859-1");
?>
```

Voir aussi [iconv_mime_decode_headers](#) , [mb_decode_mimeheader](#) , [imap_mime_header_decode](#) , [imap_base64](#) et [imap_qprint](#) .

8.49.11 iconv_mime_encode() : Construit un en-tête MIME avec les champs field_name et field_value

string iconv_mime_encode (string field_name , string field_value , array preferences)

iconv_mime_encode compose et retourne une chaîne de caractères qui représente un cahmp en-tête MIME qui ressemble à :

Subject: =?ISO-8859-1?Q?Pr=FCfung_f=FCr?= Entwerfen von einer MIME kopfzeile

Dans l'exemple ci-dessus, "Subject" est le nom du champ et la portion qui commence par "=?ISO-8859-1?..." est la valeur du champ.

Vous pouvez contrôler le comportement de la fonction iconv_mime_encode en spécifiant un tableau associatif contenant la configuration des éléments dans le paramètre preferences . La liste des éléments supportés par iconv_mime_encode figure ci-dessous. Notez que les noms des éléments sont sensibles à la casse.

Élément	Type	Description	Valeur par défaut	Exemple
scheme	boolean	Spécifie la méthode d'encodage d'un champ. Les valeurs possibles sont "B" ou "Q", où "B" signifie que le schéma d'encodage sera base64 et "Q", quoted-printable .	B	B
input-charset	string	Spécifie le jeux de caractères pour représenter le premier paramètre field_name et le second paramètre field_value . S'il est omis, iconv_mime_encode utilisera la directive de configuration iconv.internal_charset de votre php.ini pour les représenter.	iconv.internal_charset	ISO-8859-1
output-charset	string	Sépcifie le jeux de caractères à utiliser pour composer l'en-tête MIME . S'il est omis, la même valeur que input-charset sera utilisée.	le même valeur que input-charset	UTF-8
line-length	entier	Spécifie la longueur maximale de chaque en-tête. Si l'en-tête est plus grande que la longueur défini par ce paramètre, l'en-tête résultant sera un en-tête composée de plusieurs lignes conformément au standard RFC2822 - Internet Message Format . S'il est omis, la longueur maximale sera fixée à 76 caractères.	76	996
line-break-chars	string	Spécifie les caractères de fin de ligne. S'il est omis, la valeur par défaut sera "\r\n" (CR LF). Notez que ce paramètre est toujours représenté comme une chaîne ASCII au regard de la valeur du paramètre input-charset .	\r\n	\n

Exemple avec iconv_mime_encode

```
<?php
$preferences = array(
```

```

        "input-charset" => "ISO-8859-1",
        "output-charset" => "UTF-8",
        "line-length" => 76,
        "line-break-chars" => "\n"
    );
    $preferences["scheme"] = "Q";
    // Ceci affichera : "Subject: =?UTF-8?Q?Pr=C3=BCfung_Pr=C3=BCfung?="
    echo iconv_mime_encode("Subject", "Prüfung Prüfung", $preferences);

    $preferences["scheme"] = "B";
    // Ceci affichera : "Subject: =?UTF-8?B?UHLdvGZ1bmcgUHLdvGZ1bmc=?="
    echo iconv_mime_encode("Subject", "Prüfung Prüfung", $preferences);
?>

```

Voir aussi [imap_binary](#) , [mb_encode_mimeheader](#) et [imap_8bit](#) .

8.49.12 iconv_set_encoding() : Modifie le jeu courant de caractères d'encodage

bool **iconv_set_encoding** (string type , string charset)

[iconv_set_encoding](#) modifie le jeu de caractères courant, et remplace la valeur courante du paramètre type par charset . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

Les valeurs possibles de type sont :

- input_encoding
- output_encoding
- internal_encoding

Exemple avec [iconv_set_encoding](#)

```

<?php
    iconv_set_encoding("internal_encoding", "UTF-8");
    iconv_set_encoding("output_encoding", "ISO-8859-1");
?>

```

Voir aussi [iconv_get_encoding](#) et [ob_iconv_handler](#) .

8.49.13 iconv_strlen() : Retourne le nombre de caractères d'une chaîne

int **iconv_strlen** (string str , string charset)

[iconv_strlen](#) retourne le nombre de caractères de la chaîne str .

A l'opposée de [strlen](#) , la valeur de retour de [iconv_strlen](#) est le nombre de caractères faisant partie de la séquence de bytes str , ce qui n'est pas toujours la même chose que la taille en bytes de la chaîne de caractères.

Si charset n'est pas passé en paramètre, str est supposée être encodée en [iconv.internal_charset](#) .

Voir aussi [strlen](#) et [mb_strlen](#) .

8.49.14 `iconv_strpos()` : Trouve la position de la première occurrence de needle dans haystack

`int iconv_strpos (string haystack , string needle , int offset , string charset)`

`iconv_strpos` retourne la position numérique de la première occurrence de needle dans haystack .

Le paramètre optionnel `offset` spécifie la position à partir de laquelle la recherche doit commencer.

Si needle n'est pas trouvé, `iconv_strpos` retournera `FALSE` .

Attention

Cette fonction peut retourner `FALSE` , mais elle peut aussi retourner une valeur équivalent à `FALSE` , utilisable dans une condition if simple. Utilisez l' opérateur `===` pour tester la valeur de retour exacte de cette fonction.

Si haystack ou needle ne sont pas des chaînes de caractères, ils sont convertis en entiers et appliqués en tant que valeur ordinaire d'un caractère.

A l'opposé de `strpos` , la valeur retournée par `iconv_strpos` est le nombre de caractères se trouvant avant needle, plutôt que la position en octets où needle a été trouvé. Les caractères sont comptés en se basant sur le jeu de caractères spécifié par `charset` .

Si le paramètre `charset` est omis, string sera encodé en accord avec `iconv.internal_charset` .

Voir aussi `strpos` , `mb_strpos` et `iconv_strrpos` .

8.49.15 `iconv_strrpos()` : Trouve la position de la dernière occurrence d'un élément dans une chaîne, à partir d'un offset

`int iconv_strrpos (string haystack , string needle , string charset)`

`iconv_strrpos` retourne la position numérique de la dernière occurrence de needle dans haystack .

Si needle n'est pas trouvé, `iconv_strrpos` retournera `FALSE` .

Attention

Cette fonction peut retourner `FALSE` , mais elle peut aussi retourner une valeur équivalent à `FALSE` , utilisable dans une condition if simple. Utilisez l' opérateur `===` pour tester la valeur de retour exacte de cette fonction.

Si haystack ou needle ne sont pas des chaînes de caractères, ils seront convertis en chaîne de caractères et reconnus comme code ASCII de chaque caractères.

A l'opposé de la fonction `strpos` , la valeur retournée par `iconv_strrpos` est le nombre de caractères avant needle, plutôt que la position en octets de needle. Les caractères sont comptés en se basant sur le jeu de caractères `charset` .

Voir aussi `strrpos` , `iconv_strpos` et `mb_strrpos` .

8.49.16 iconv_substr() : Retourne une partie de chaîne iconv

string **iconv_substr** (string *str* , int *offset* , int *length* , string *charset*)

iconv_substr retourne le segment de *str* défini par *start* et *length* .

Si *start* est positif, la chaîne retournée commencera au caractère numéro *start* , dans la chaîne *str* . Le premier caractère est numéroté zéro.

Si *start* est négatif, la chaîne retournée commencera au caractère numéro *start* dans la chaîne *str* .

Si *length* est fourni et positif, la chaîne retournée contiendra au plus *length* caractères, en commençant à partir du caractère *start* (en fonction de la chaîne *str*). Si *str* est plus petite que *start* , iconv_substr retournera FALSE .

Si *length* est fourni et négatif, alors le même nombre de caractères sera omis, en partant de la fin de la chaîne *str* . Si *start* représente une position hors de la chaîne, une chaîne vide sera retournée.

Notez que *offset* et *length* sont toujours considérés comme représentant une position calculée sur la représentation ASCII des caractères déterminés par *charset* , contrairement à substr qui se base toujours sur la position par octet. Si *charset* est omis, le jeu de caractères est déterminé par iconv.internal_encoding .

Voir aussi substr , mb_substr et mb_strcut .

8.49.17 iconv() : Convertit une chaîne dans un jeu de caractères

string **iconv** (string *in_charset* , string *out_charset* , string *str*)

iconv convertit la chaîne *str* depuis le jeu de caractères *in_charset* vers le jeu de caractères *out_charset* . Elle retourne la chaîne ainsi convertie, ou bien FALSE , en cas d'échec.

Si vous ajoutez la chaîne //TRANSLIT au paramètre *out_charset* , la translittération est activée. Cela signifie que lorsqu'un caractère ne peut être représenté dans le jeu de caractères cible, il peut être représenté approximativement à partir d'un ou plusieurs caractères représentant le même caractère. Si vous ajoutez la chaîne //IGNORE , les caractères qui ne peuvent être représentés dans le jeu de caractères cible sont tout simplement ignorés. Sinon, *str* sera coupé à partir du premier caractère illégal rencontré.

Exemple avec iconv

```
<?php
echo iconv("ISO-8859-1","UTF-8","Ceci est un test.");
?>
```

8.49.18 ob_iconv_handler() : Gestionnaire de sortie pour maîtriser le jeu de caractères de sortie

string **ob_iconv_handler** (string *contents* , int *status*)

ob_iconv_handler convertit la chaîne utilisant le jeu de caractères *internal_encoding* en une chaîne utilisant celui de *output_encoding* .

`internal_encoding` et `output_encoding` doivent être définis par `iconv_set_encoding` ou dans le fichier de configuration `php.ini` .

Exemple avec `ob_iconv_handler`

```
<?php
    ob_start("ob_iconv_handler"); // start output buffering
?>
```

Voir aussi `iconv_get_encoding` et `iconv_set_encoding` , et `bufferisation de sortie` .

8.50 Fonctions ID3

8.50.1 Introduction

Ces fonctions vous permettent de lire et de manipuler les tags ID3. Les tags ID3 sont utilisés dans les fichiers MP3 pour stocker le titre d'une chanson, tout comme des informations sur l'artiste, l'album, le genre, l'année et le numéro de la piste.

Depuis la version 0.2, il est également possible d'extraire des champs texte depuis des tags ID3 v2.2+.

8.50.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.50.3 Installation

id3 est une partie de PECL et peut être installé en utilisant l'installateur Pear. Pour compiler PHP avec le support ID3, téléchargez le code source, mettez-le dans le dossier php-src/ext/id3 et compilez en spécifiant `--enable-id3`.

8.50.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.50.5 Types de ressources

Cette extension ne définit aucune ressource.

8.50.6 Constantes pré-définies

La plupart des fonctions ID3 vous permet soit de spécifier, soit de retourner un tag. Pour spécifier la version, utilisez une de ces constantes :

`ID3_V1_0` (entier)

`ID3_V1_0` est utilisé si vous travaillez avec des tags ID3 V1.0. Ces tags peuvent contenir les champs `title` , `artist` , `album` , `genre` , `year` et `comment` .

`ID3_V1_1` (entier)

`ID3_V1_1` est utilisé si vous travaillez avec des tags ID3 V1.1. Ces tags peuvent contenir tous les champs de la version 1.0 ainsi qu'un champ représentant le numéro de la piste.

`ID3_V2_1` (entier)

`ID3_V2_1` est utilisé si vous travaillez avec des tags ID3 V2.1.

`ID3_V2_2` (entier)

`ID3_V2_2` est utilisé si vous travaillez avec des tags ID3 V2.2 tags.

`ID3_V2_3` (entier)

ID3_V2_3 est utilisé si vous travaillez avec des tags ID3 V2.3 tags.

ID3_V2_4 (entier)

ID3_V2_4 est utilisé si vous travaillez avec des tags ID3 V2.4 tags.

ID3_BEST (entier)

ID3_BEST est utilisé si vous voulez laisser les fonctions id3 déterminer quelle version de tags doit être utilisée.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [id3_get_frame_long_name](#)
- [id3_get_frame_short_name](#)
- [id3_get_genre_id](#)
- [id3_get_genre_list](#)
- [id3_get_genre_name](#)
- [id3_get_tag](#)
- [id3_get_version](#)
- [id3_remove_tag](#)
- [id3_set_tag](#)

8.50.8 id3_get_frame_short_name() : Récupère le nom court d'un champs ID3v2

string **id3_get_frame_short_name** (string frameld)

[id3_get_frame_short_name](#) retourne le nom court d'un champs ID3v2.

Exemple avec [id3_get_frame_short_name](#)

```
<?php
$shortName = id3_get_frame_short_name("TOLY");
echo $shortName;
?>
```

L'exemple ci-dessus va afficher :

originalLyricist

Les valeurs retournées par [id3_get_short_name](#) sont utilisées dans le tableau retourné par [id3_get_tag](#) .

Voir aussi [id3_get_frame_long_name](#) .

8.50.9 id3_get_genre_id() : Récupération d'un id pour un genre

int **id3_get_genre_id** (string genre)

[id3_get_genre_id](#) retourne l'id pour un genre. Si le genre spécifié n'est pas disponible dans la liste des genres, [id3_get_genre_id](#) retournera FALSE .

Dans un tag ID3, le genre est stocké en utilisant un entier compris dans l'intervale 0-147.

Exemple avec `id3_get_genre_id`

```
<?php
$id = id3_get_genre_id("Alternative");
echo $id;
?>
```

L'exemple ci-dessus va afficher :

20

Voir aussi `id3_get_genre_list` et `id3_get_genre_name` .

8.50.10 `id3_get_genre_list()` : Récupère toutes les valeurs possibles du genre

array `id3_get_genre_list ()`

`id3_get_genre_list` retourne un tableau contenant tous les genres possibles pouvant être stockés dans un tag ID3. La liste a été créée par Eric Kemp et, plus tard, complétée par WinAmp.

Cette fonction est pratique pour fournir à vos utilisateurs une liste de genres depuis laquelle ils peuvent en choisir un. Lorsque vous mettez à jour vos tags ID3, vous devez également spécifier un genre sous forme d'entier entre 0-147.

Exemple avec `id3_get_genre_list`

```
<?php
$genres = id3_get_genre_list();
print_r($genres);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[0] => Blues
[1] => Classic Rock
[2] => Country
[3] => Dance
[4] => Disco
[5] => Funk
[6] => Grunge
[7] => Hip-Hop
[8] => Jazz
[9] => Metal
[10] => New Age
[11] => Oldies
[12] => Other
[13] => Pop
[14] => R&B
```

[15] => Rap
[16] => Reggae
[17] => Rock
[18] => Techno
[19] => Industrial
[20] => Alternative
[21] => Ska
[22] => Death Metal
[23] => Pranks
[24] => Soundtrack
[25] => Euro-Techno
[26] => Ambient
[27] => Trip-Hop
[28] => Vocal
[29] => Jazz+Funk
[30] => Fusion
[31] => Trance
[32] => Classical
[33] => Instrumental
[34] => Acid
[35] => House
[36] => Game
[37] => Sound Clip
[38] => Gospel
[39] => Noise
[40] => Alternative Rock
[41] => Bass
[42] => Soul
[43] => Punk
[44] => Space
[45] => Meditative
[46] => Instrumental Pop
[47] => Instrumental Rock
[48] => Ethnic
[49] => Gothic
[50] => Darkwave
[51] => Techno-Industrial
[52] => Electronic
[53] => Pop-Folk
[54] => Eurodance
[55] => Dream
[56] => Southern Rock
[57] => Comedy
[58] => Cult
[59] => Gangsta
[60] => Top 40
[61] => Christian Rap
[62] => Pop/Funk
[63] => Jungle
[64] => Native US
[65] => Cabaret
[66] => New Wave
[67] => Psychadelic
[68] => Rave

[69] => Showtunes
[70] => Trailer
[71] => Lo-Fi
[72] => Tribal
[73] => Acid Punk
[74] => Acid Jazz
[75] => Polka
[76] => Retro
[77] => Musical
[78] => Rock & Roll
[79] => Hard Rock
[80] => Folk
[81] => Folk-Rock
[82] => National Folk
[83] => Swing
[84] => Fast Fusion
[85] => Bebob
[86] => Latin
[87] => Revival
[88] => Celtic
[89] => Bluegrass
[90] => Avantgarde
[91] => Gothic Rock
[92] => Progressive Rock
[93] => Psychedelic Rock
[94] => Symphonic Rock
[95] => Slow Rock
[96] => Big Band
[97] => Chorus
[98] => Easy Listening
[99] => Acoustic
[100] => Humour
[101] => Speech
[102] => Chanson
[103] => Opera
[104] => Chamber Music
[105] => Sonata
[106] => Symphony
[107] => Booty Bass
[108] => Primus
[109] => Porn Groove
[110] => Satire
[111] => Slow Jam
[112] => Club
[113] => Tango
[114] => Samba
[115] => Folklore
[116] => Ballad
[117] => Power Ballad
[118] => Rhythmic Soul
[119] => Freestyle
[120] => Duet
[121] => Punk Rock
[122] => Drum Solo

```
[123] => Acapella
[124] => Euro-House
[125] => Dance Hall
[126] => Goa
[127] => Drum & Bass
[128] => Club-House
[129] => Hardcore
[130] => Terror
[131] => Indie
[132] => BritPop
[133] => Negerpunk
[134] => Polsk Punk
[135] => Beat
[136] => Christian Gangsta
[137] => Heavy Metal
[138] => Black Metal
[139] => Crossover
[140] => Contemporary C
[141] => Christian Rock
[142] => Merengue
[143] => Salsa
[144] => Thrash Metal
[145] => Anime
[146] => JPop
[147] => SynthPop
)
```

Voir aussi [id3_get_genre_name](#) et [id3_get_genre_id](#) .

8.50.11 id3_get_genre_name() : Récupère le nom pour un id de genre

string [id3_get_genre_name](#) (int genre_id)

[id3_get_genre_name](#) retourne le nom pour le genre dont l'identifiant est genre_id .

Dans un tag ID3, le genre est stocké en utilisant un entier compris entre 0 et 147.

Exemple avec [id3_get_genre_name](#)

```
<?php
$genre = id3_get_genre_name(20);
echo $genre;
?>
```

L'exemple ci-dessus va afficher :

Alternative

Voir aussi [id3_get_genre_list](#) et [id3_get_genre_id](#) .

8.50.12 id3_get_tag() : Récupère toutes les informations stockées dans un tag ID3

array **id3_get_tag** (string filename , int version)

`id3_get_tag` est utilisée pour récupérer toutes les informations stockées dans un tag ID3 d'un fichier spécifique.

Note

Au lieu d'un nom de fichier, vous pouvez également passer une ressource valide.

Le paramètre optionnel version vous permet de spécifier la version du tag, bien que les fichiers MP3 doivent contenir à la fois les versions 1.x et les versions 2.x.

Exemple avec `id3_get_tag`

```
<?php
$tag = id3_get_tag( "chemin/vers/exemple.mp3" );
print_r($tag);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [title] => DN-38416
    [artist] => Re:\Legion
    [album] => Reflections
    [year] => 2004
    [genre] => 19
)
```

La clé genre doit contenir un entier compris entre 0 et 147. Vous pouvez utiliser la fonction `id3_get_genre_name` pour convertir ce nombre en une chaîne lisible humainement.

Depuis la version 0.2, `id3_get_tag` supporte également les tags ID3 version 2.2, 2.3 et 2.4. Pour extraire les informations de ces tags, passez une des constantes `ID3_V2_2`, `ID3_V2_3` ou `ID3_V2_4` en tant que second paramètre.

Exemple avec `id3_get_tag`

```
<?php
$tag = id3_get_tag( "chemin/vers/exemple2.mp3", ID3_V2_3 );
print_r($tag);
?>
```

L'exemple ci-dessus va afficher quelque chose de similaire à :

```
Array
(
    [copyright] => Dirty Mac
    [originalArtist] => Dirty Mac
    [composer] => Marcus Götze
    [artist] => Dirty Mac
)
```

```
[title] => Little Big Man
[album] => Demo-Tape
[track] => 5/12
[genre] => (17)Rock
[year] => 2001
)
```

Les tags ID3 v2.x peuvent contenir beaucoup plus d'informations concernant un fichier MP3 par rapport aux tags ID3 version 1.x.

Voir aussi [id3_set_tag](#) , [id3_remove_tag](#) et [id3_get_version](#) .

8.50.13 id3_get_version() : Récupère la version d'un tag ID3

int **id3_get_version** (string filename)

[id3_get_version](#) retourne la(les) version(s) du(des) tag(s) ID3 d'un fichier MP3. Sachant qu'un tag peut contenir des tags ID3 de version v1.x et v2.x, la valeur retournée par cette fonction doit être comparée au niveau binaire avec les constantes pré-définies ID3_V1_0 , ID3_V1_1 et ID3_V2 .

Note

Au lieu d'un nom de fichier, vous pouvez également passer une ressource valide.

Exemple avec [id3_get_version](#)

```
<?php
$version = id3_get_version("chemin/vers/exemple.mp3");
if ($version & ID3_V1_0) {
    echo "Contient un tag version 1.x\n";
}
if ($version & ID3_V1_1) {
    echo "Contient un tag version 1.1\n";
}
if ($version & ID3_V2) {
    echo "Contient un tag version 2.x\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
Contient un tag version 1.x
Contient un tag version 1.1
```

Si un fichier contient un tag ID3 v1.1, il contient toujours un tag 1.0, vu que la version 1.1 n'est qu'une extension de la version 1.0.

Voir aussi [id3_get_tag](#) , [id3_set_tag](#) et [id3_remove_tag](#) .

8.50.14 id3_remove_tag() : Efface un tag ID3

bool **id3_remove_tag** (string filename , int version)

[id3_remove_tag](#) est utilisé pour effacer les informations stockées dans un tag ID3. Si aucun tag n'est présent, [id3_remove_tag](#) retournera FALSE et quitte le fichier tel que.

Note

Au lieu d'un nom de fichier, vous pouvez également passer une ressource valide.

Le paramètre optionnel `version` vous permet de spécifier la version du tag, bien que les fichiers MP3 doivent contenir à la fois les versions 1.x et les versions 2.x.

Exemple avec `id3_remove_tag`

```
<?php
$result = id3_remove_tag( "chemin/vers/exemple.mp3", ID3_V1_0 );
if ( $result === true ) {
    echo "Le tag a été effacé avec succès\n";
}
?>
```

Si le fichier est accessible en écriture et qu'il contient un tag 1.0, l'exemple affichera :

Le tag a été effacé avec succès

Note

Actuellement, `id3_remove_tag` supporte uniquement les tags version 1.0 et 1.1. Si vous tentez d'effacer un tag 1.0 et que le fichier contient un tag 1.1, le tag sera effacé car la version 1.1 n'est qu'une extension de la version 1.0.

Voir aussi `id3_get_tag` , `id3_set_tag` et `id3_get_version` .

8.50.15 `id3_set_tag()` : Met à jour les informations stockées dans un tag ID3

bool `id3_set_tag` (string `filename` , array `tag` , int `version`)

`id3_set_tag` est utilisé pour changer les informations stockées dans un tag ID3. Si aucun tag n'est présent, le tag sera ajouté au fichier.

Note

Au lieu d'un nom de fichier, vous pouvez également passer une ressource valide.

Le paramètre optionnel `version` vous permet de spécifier la version du tag, bien que les fichiers MP3 doivent contenir à la fois les versions 1.x et les versions 2.x.

Exemple avec `id3_set_tag`

```
<?php
$data = array(
    "title" => "Re:Start",
    "artist" => "Re:\Legion",
    "comment" => "A nice track"
);
$result = id3_set_tag( "path/to/example.mp3", $data, ID3_V1_0 );
if ( $result === true ) {
    echo "Le tag a été mis à jour avec succès\n";
}
?>
```

Si le fichier est accessible en écriture, l'exemple affichera :

Le tag a été mis à jour avec succès

Note

Actuellement, `id3_set_tag` supporte uniquement les versions 1.0 et 1.1.

Les clés suivantes peuvent être utilisées dans le tableau associatif :

Clé	Valeur possible	disponible dans la version
title	chaîne de caractères avec un maximum de 30 caractères	v1.0, v1.1
artist	chaîne de caractères avec un maximum de 30 caractères	v1.0, v1.1
album	chaîne de caractères avec un maximum de 30 caractères	v1.0, v1.1
year	4 chiffres	v1.0, v1.1
genre	valeur entière comprise entre 0 et 147	v1.0, v1.1
comment	chaîne de caractères avec un maximum de 30 caractères (28 en v1.1)	v1.0, v1.1
track	entier compris entre 0 et 255	v1.1

Voir aussi `id3_get_tag` , `id3_remove_tag` et `id3_get_version` .

8.51 Informix

8.51.1 Introduction

Les pilotes d'accès à Informix pour Online (ODS) 7.x, SE 7.x, Universal Server (IUS) 9.x et IDS 2000 sont implémentés dans "functions/ifx.ec" et "functions/php3_ifx.h". Le support ODS 7.x est plutôt complet, et accepte les colonnes de type BYTE et TEXT. Le support IUS 9.x est partiellement fini, de nouveaux types sont disponibles, mais SLOB et CLOB sont toujours en cours de développement.

8.51.2 Pré-requis

Note

Configuration notes

Vous avez besoin d'une version de ESQL/C pour compiler le pilote PHP d'Informix. Les versions ESQL/C 7.2x sont utilisables. ESQL/C fait partie du SDK Informix Client.

Avant que vous ne lanciez le script "configure", assurez-vous que la variable d'environnement "INFORMIXDIR" a été correctement paramétrée, et que \$INFORMIXDIR/bin est dans votre PATH.

8.51.3 Installation

Pour activer ces fonctions, vous devez compiler PHP avec l'option `--with-informix[=DIR]`, où DIR est le dossier d'installation de Informix et, par défaut, il ne vaut rien.

8.51.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

Note

Avant que vous ne lanciez le script "configure", assurez-vous que la variable d'environnement "INFORMIXDIR" a été correctement paramétrée, et que \$INFORMIXDIR/bin est dans votre PATH. Vérifiez cela en étudiant le résultat de la fonction `phpinfo` avant de commencer à utiliser cette extension. `phpinfo` devrait lister ces variables d'environnement. C'est le cas aussi bien pour les modes CGI que module Apache. Vous pouvez aussi mettre en place ces variables dans votre script de démarrage Apache.

Les bibliothèques partagées Informix doivent être accessibles pour le chargement (vérifiez `LD_LIBRARY_PATH` et `ld.so.conf/ldconfig`).

Note

Notes sur l'utilisation des BLOBs (TEXT et BYTE)

Les objets de type BLOB sont normalement gérés par des identifiants de BLOB. Les requêtes de sélection retournent un identifiant de BLOB pour chaque colonne de type BYTE et TEXT. Vous pouvez en lire le contenu, avec des commandes de type `"string_var = ifx_get_blob($BLOB_id);"` ; si vous souhaitez ramener le BLOB en mémoire (avec: `"ifx_blobinfile_mode(0);"`). Si vous préférez recevoir le contenu d'une colonne BLOB dans un fichier, utilisez `ifx_blobinfile_mode`, et `ifx_get_blob($BLOB_id)` vous retournera le nom du fichier. Utilisez les fonctions habituelles d'accès

aux fichiers pour lire son contenu.

Pour les requêtes INSERT/UPDATE, vous devez créer les identifiants de BLOB vous-même, avec la fonction `ifx_create_blob`. Puis, vous placez l'identifiant de BLOB dans un tableau, et remplacez la colonne par un point d'interrogation (?). Pour les UPDATE/INSERT, vous êtes responsable du contenu du BLOB, avec la fonction `ifx_update_blob`.

Le comportement par défaut des colonnes de type BLOB peut être modifié en affectant de nouvelles valeurs aux variables de configuration (même à la volée) :

Variable de configuration : `ifx.textasvarchar`

Variable de configuration : `ifx.byteasvarchar`

Fonctions à utiliser lors de l'exécution :

`ifx_textasvarchar(0)` : utilise l'identifiant de BLOB avec des colonnes de type TEXT, dans les requêtes SELECT

`ifx_byteasvarchar(0)` : utilise l'identifiant de BLOB avec des colonnes de type BYTE, dans les requêtes SELECT

`ifx_textasvarchar(1)` : retourne les colonnes de type TEXT sous la forme de VARCHAR, sans utiliser les identifiants de BLOB dans les requêtes SELECT.

`ifx_byteasvarchar(1)` : retourne les colonnes de type BYTE sous la forme de VARCHAR, sans utiliser les identifiants de BLOB dans les requêtes SELECT.

Variable de configuration : `ifx.BLOBinfile`

Fonctions à utiliser lors de l'exécution :

`ifx_blobinfile_mode(0)` : retourne les colonnes de type BYTE en mémoire ; l'identifiant de BLOB vous donnera accès au contenu.

`ifx_blobinfile_mode(1)` : retourne les colonnes de type BYTE dans un fichier ; l'identifiant de BLOB vous donnera accès au nom de ce fichier.

En affectant la valeur 1 à `ifx_text/byteasvarchar`, vous pouvez utiliser les colonnes de type TEXT et BYTE dans les requêtes SELECT comme des champs VARCHAR (mais plus long). Etant donné la gestion des chaînes par PHP, cette technique conserve les données binaires. Les données retournées peuvent contenir n'importe quoi, et vous êtes responsable de la bonne manipulation de ces valeurs.

En affectant la valeur 1 à `ifx_blobinfile_mode`, utilisez le nom de fichier retourné par `ifx_get_blob` pour accéder au contenu du BLOB. Notez bien que vous êtes tenu responsable de l'effacement des fichiers temporaires, créés par Informix. Chaque nouvelle ligne lue sur le serveur va créer un nouveau fichier temporaire, pour chaque colonne de type BYTE.

L'emplacement des fichiers temporaires peut être modifié, grâce à la variable "blobdir", (par défaut, ".", c'est-à-dire, le dossier courant). Une valeur telle que `BLOBdir="tmpBLOB"` simplifiera le nettoyage des fichiers temporaires, accidentellement oubliés (les noms commencent tous par "blb").

Note

Suppression automatique des espaces (SQLCHAR et SQLNCHAR)

Elle peut être mise en place avec la variable de configuration.

ifx.charasvarchar : avec la valeur 1, les espaces de fin de champs seront automatiquement supprimés.

Note

NULL values

Lorsque la variable de configuration ifx.nullformat (ou que la fonction `ifx_nullformat`) est à un, les colonnes contenant la valeur NULL retourneront la chaîne "NULL" et, sinon, retourneront une chaîne vide. Cela vous permet de faire la différence entre les colonnes vides et celles qui contiennent la valeur NULL.

Nom	Par défaut	Modifiable	Historique
ifx.allow_persistent	"1"	PHP_INI_SYSTEM	
ifx.max_persistent	"-1"	PHP_INI_SYSTEM	
ifx.max_links	"-1"	PHP_INI_SYSTEM	
ifx.default_host	NULL	PHP_INI_SYSTEM	
ifx.default_user	NULL	PHP_INI_SYSTEM	
ifx.default_password	NULL	PHP_INI_SYSTEM	
ifx.blobinfile	"1"	PHP_INI_ALL	
ifx.textasvarchar	"0"	PHP_INI_ALL	
ifx.byteasvarchar	"0"	PHP_INI_ALL	
ifx.charasvarchar	"0"	PHP_INI_ALL	
ifx.nullformat	"0"	PHP_INI_ALL	

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [Directives du php.ini](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

ifx.allow_persistent bool

Active les connexions persistantes à une base de données Informix.

ifx.max_persistent entier

Nombre maximum de connexions persistantes à une base de données Informix, par processus.

ifx.max_links entier

Nombre maximum de connexions à une base de données Informix par processus, en incluant les connexions persistantes.

ifx.default_host string

Hôte par défaut où se connecter si aucun hôte n'est spécifié par les fonctions `ifx_connect` ou `ifx_pconnect`. Cette option ne s'applique pas si le safe mode est activé.

ifx.default_user string

Utilisateur par défaut si aucun utilisateur n'est spécifié par les fonctions `ifx_connect` ou `ifx_pconnect`. Cette option ne s'applique pas si le safe mode est activé.

ifx.default_password string

Mot de passe par défaut si aucun mot de passe n'est spécifié par les fonctions `ifx_connect` ou `ifx_pconnect`. Cette option ne s'applique pas si le safe mode est activé.

ifx.blobinfile bool

Lorsque cette option est activée, les colonnes de type "blob" seront retournées dans un fichier. Par défaut, elles seront retournées en mémoire. Il est possible de modifier dynamiquement cette valeur grâce à la fonction [ifx_blobinfile_mode](#) .

ifx.textasvarchar bool

Lorsque cette option est activée, les colonnes de type "TEXT" seront retournées dans une chaîne de caractères. Par défaut, elles seront retournées en mémoire. Il est possible de modifier dynamiquement cette valeur grâce à la fonction [ifx_textasvarchar](#) .

ifx.byteasvarchar bool

Lorsque cette option est activée, les colonnes de type "BYTE" seront retournées dans une chaîne de caractères. Par défaut, elles seront retournées en mémoire. Il est possible de modifier dynamiquement cette valeur grâce à la fonction [ifx_textasvarchar](#) .

ifx.charasvarchar bool

Lorsque cette option est activée, les espaces en fin de chaîne de caractères seront conservés lors d'une commande FETCH.

ifx.nullformat bool

Lorsque cette option est activée, les colonnes de valeur NULL seront retournées comme des chaînes de caractères vides. Il est possible de modifier dynamiquement cette valeur grâce à la fonction [ifx_nullformat](#) .

8.51.5 Types de ressources

8.51.6 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [ifx_affected_rows](#)
- [ifx_blobinfile_mode](#)
- [ifx_byteasvarchar](#)
- [ifx_close](#)
- [ifx_connect](#)
- [ifx_copy_blob](#)
- [ifx_create_blob](#)
- [ifx_create_char](#)
- [ifx_do](#)
- [ifx_error](#)
- [ifx_errormsg](#)
- [ifx_fetch_row](#)
- [ifx_fieldproperties](#)
- [ifx_fieldtypes](#)
- [ifx_free_blob](#)
- [ifx_free_char](#)
- [ifx_free_result](#)
- [ifx_get_blob](#)
- [ifx_get_char](#)

- [ifx_getsqlca](#)
- [ifx_htmltbl_result](#)
- [ifx_nullformat](#)
- [ifx_num_fields](#)
- [ifx_num_rows](#)
- [ifx_pconnect](#)
- [ifx_prepare](#)
- [ifx_query](#)
- [ifx_textasvarchar](#)
- [ifx_update_blob](#)
- [ifx_update_char](#)
- [ifxus_close_slob](#)
- [ifxus_create_slob](#)
- [ifxus_free_slob](#)
- [ifxus_open_slob](#)
- [ifxus_read_slob](#)
- [ifxus_seek_slob](#)
- [ifxus_tell_slob](#)
- [ifxus_write_slob](#)

8.51.8 ifx_blobinfile_mode() : Choisit le mode par défaut des objets BLOB pour toutes les requêtes SELECT

void **ifx_blobinfile_mode** (int mode)

[ifx_blobinfile_mode](#) modifie le mode par défaut des objets BLOB pour toutes les requêtes SELECT. Mode "0" chargera les BLOB de type Byte en mémoire ; Mode "1" sauvera les BLOB de type Byte dans un fichier.

8.51.9 ifx_byteasvarchar() : Choisit le mode par défaut des objets BYTE

void **ifx_byteasvarchar** (int mode)

[ifx_byteasvarchar](#) modifie le mode par défaut des objets BYTE. Le mode "0" retournera l'identifiant de BLOB, et le mode "1" retournera le contenu du TEXT sous la forme d'un VARCHAR.

8.51.10 ifx_close() : Ferme une connexion à un serveur Informix

int **ifx_close** (int link_identifiant)

[ifx_close](#) retourne toujours TRUE .

[ifx_close](#) ferme le lien au serveur de données Informix associé à l'identifiant de connexion link_identifiant . Si l'identifiant du lien n'est pas spécifié, la dernière connexion est utilisée.

Notez qu'il n'est généralement pas besoin d'appeler cette fonction, car les connexions non persistantes seront automatiquement fermées.

[ifx_close](#) ne peut pas fermer une connexion ouverte avec [ifx_pconnect](#) .

Fermer une connexion Informix

```
<?php
$conn_id = ifx_connect ("mydb@ol_srv", "coucou", "cestmoi");
//... quelques requêtes diverses et variées ...
ifx_close($conn_id);
?>
```

Voir aussi [ifx_connect](#) et [ifx_pconnect](#) .

8.51.11 ifx_connect() : Ouvre une connexion à un serveur Informix

int **ifx_connect** (*string database* , *string userid* , *string password*)

[ifx_connect](#) retourne un identifiant de connexion, en cas de succès, et FALSE sinon.

[ifx_connect](#) établit une connexion à un serveur Informix. Tous les arguments sont optionnels, et, s'ils viennent à manquer, les valeurs par défaut seront prises dans le fichier [de configuration](#) . (ifx.default_host pour l'hôte par défaut) (Les bibliothèques Informix utiliseront la variable d'environnement **\$INFORMIXSERVER** si ifx.default_host n'est pas définie). ifx.default_user pour l'utilisateur, et ifx.default_password comme mot de passe (si aucun n'a été défini).

Si un deuxième appel à [ifx_connect](#) est fait avec les mêmes arguments, l'identifiant de connexion déjà ouvert sera retourné.

Le lien avec le serveur sera fermé dès que le script se termine, ce qui fait qu'il n'est pas nécessaire de terminer les connexions avec [ifx_close](#) .

Connexion à un serveur Informix

```
<?php
$conn_id = ifx_connect ("mydb@ol_srv1", "imyself", "mypassword");
?>
```

Voir aussi [ifx_pconnect](#) et [ifx_close](#) .

8.51.12 ifx_copy_blob() : Duplique un objet BLOB

int **ifx_copy_blob** (int bid)

Duplique un objet BLOB dont l'identifiant est bid .

[ifx_copy_blob](#) retourne FALSE en cas d'erreur ou l'identifiant du nouvel objet en cas de succès.

8.51.13 ifx_create_blob() : Crée un objet BLOB

int **ifx_create_blob** (int type , int mode , string param)

[ifx_create_blob](#) crée un objet BLOB.

type: 1 = TEXT, 0 = BYTE

mode: 0 = L'objet BLOB place le contenu en mémoire ; 1 = L'objet BLOB place le contenu dans un fichier.

param: Si mode = 0: pointeur du contenu, si mode = 1: pointeur vers un fichier.

ifx_create_blob retourne FALSE en cas d'erreur, et sinon, un identifiant de BLOB.

8.51.14 ifx_create_char() : Crée un objet char

int **ifx_create_char** (string param)

ifx_create_char crée un objet char. param sera le contenu de l'objet.

8.51.15 ifx_do() : Exécute une requête SQL préparée

int **ifx_do** (int result_id)

ifx_do exécute une requête qui a déjà été préparée, ou crée un pointeur pour cela.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

ifx_do ne libère pas result_id en cas d'erreur.

De plus, ifx_do fixe la valeur de result_id pour accès ultérieur par ifx_affected_rows .

Voir aussi ifx_prepare .

8.51.16 ifx_error() : Retourne le code d'erreur de la dernière requête Informix

string **ifx_error** (void)

ifx_error retourne le code d'erreur de la dernière requête Informix. Les codes d'erreur Informix (SQLSTATE et SQLCODE) sont formatés comme suit :

x [SQLSTATE = aa bbb SQLCODE=cccc]

avec x = space : aucune erreur

E : erreur

N : il n'y a plus d'informations

W : alerte

? : indéfinie

Si le caractère vaut autre chose qu'un espace, SQLSTATE et SQLCODE décrivent l'erreur avec plus de détails.

Reportez-vous au manuel Informix pour trouver la description de SQLSTATE et SQLCODE

ifx_error retourne une chaîne avec un caractère, décrivant le résultat général de la commande, et

aussi SQLSTATE et SQLCODE associés à la plus récente requête SQL exécutée. Le format de la chaîne est "(char) [SQLSTATE=(deux chiffres) (trois chiffres) SQLCODE=(un chiffre)]". Le premier caractère peut être ' ' (espace) (succès), ' W ' (Alerte), ' E ' (une erreur est survenue durant le traitement) ou ' N ' (aucune donnée de retour).

Voir aussi [ifx_errormsg](#) .

8.51.17 ifx_errormsg() : Retourne le message d'erreur de la dernière requête Informix

string **ifx_errormsg** (*int* *errorcode*)

[ifx_errormsg](#) retourne le plus récent message d'erreur ou, lorsque l'option *errorcode* est présente, le message d'erreur associé à *errorcode* .

Exemple avec [ifx_errormsg](#)

```
<?php
printf("%s\n<br />", ifx_errormsg(-201));
?>
```

Voir aussi [ifx_error](#) .

8.51.18 ifx_fetch_row() : Retourne une ligne sous la forme d'un tableau énuméré

array **ifx_fetch_row** (*int* *result_id* , *mixed* *position*)

[ifx_fetch_row](#) retourne un tableau associatif qui contient la ligne retournée ou FALSE s'il ne reste plus de lignes à lire, ou s'il a eu une erreur.

Les colonnes de types BLOB sont retournées sous la forme d'un identifiant à utiliser avec [ifx_get_blob](#) à moins que vous n'ayez utilisé la fonction [ifx_textasvarchar](#) ou [ifx_byteasvarchar](#) et, dans ce cas, les BLOBs seront retournés sous forme de chaîne. [ifx_fetch_row](#) retourne FALSE en cas d'erreur.

result_id est un identifiant valide de résultat, retourné par [ifx_query](#) ou [ifx_prepare](#) (Requêtes SELECT seulement !).

position est un paramètre optionnel, pour une opération de lecture d'informations sur un pointeur de type "scroll": "NEXT", "PREVIOUS", "CURRENT", "FIRST", "LAST" ou encore un nombre. Si vous spécifiez un nombre, la ligne d'index absolu sera retournée. Ce paramètre est optionnel, et ne fonctionne qu'avec les pointeurs de type "scroll".

[ifx_fetch_row](#) retourne une ligne de données d'un résultat associé à l'identifiant de résultat *result_id* . La ligne est retournée sous la forme d'un tableau associatif. Chaque résultat est stocké dans un tableau associatif, dont l'index numérique commence à 0 et ayant comme clé, le nom du champ.

Les appels ultérieurs à [ifx_fetch_row](#) retourneront la ligne suivante, ou FALSE s'il n'y a plus de ligne.

Exemple avec [ifx_fetch_row](#)

```

<?php
$rid = ifx_prepare ("select * from emp where name like " . $name,
                  $connid, IFX_SCROLL);
if (! $rid) {
//   ... erreur ...
}
$rowcount = ifx_affected_rows($rid);
if ($rowcount > 1000) {
    printf ("Trop de lignes dans le résultat. (%d)\n<br />", $rowcount);
    die ("Recommencez votre requête. <br />\n");
}
if (! ifx_do ($rid)) {
//   ... erreur ...
}
$row = ifx_fetch_row ($rid, "NEXT");
while (is_array($row)) {
    for (reset($row); $fieldname=key($row); next($row)) {
        $fieldvalue = $row[$fieldname];
        printf ("%s = %s,", $fieldname, $fieldvalue);
    }
    printf("\n<br />");
    $row = ifx_fetch_row($rid, "NEXT");
}
ifx_free_result ($rid);
?>

```

8.51.19 ifx_fieldproperties() : Liste les propriétés des champs SQL

array **ifx_fieldproperties** (int result_id)

ifx_fieldproperties retourne un tableau associatif avec les noms des champs comme clés, et les données de propriétés des champs comme valeurs. ifx_fieldproperties retourne FALSE en cas d'erreur.

ifx_fieldproperties retourne les propriétés Informix SQL pour tous les champs d'une requête, sous la forme d'un tableau associatif. Les propriétés sont présentées sous la forme : "SQLTYPE;longueur;précision;échelle;ISNULLABLE" avec SQLTYPE qui représente le type de données Informix tel que "SQLVCHAR" et ISNULLABLE = "Y" ou "N" (le champ peut contenir NULL ou pas : Oui ou Non).

Exemple avec ifx_fieldproperties

```

$properties = ifx_fieldproperties($resultid);
if (!isset($properties)) {
    /* ... erreur ... */
}
foreach ($properties as $fname => $val) {
    echo "$fname:\t property = $val\n";
}
?>

```

8.51.20 ifx_fieldtypes() : Liste les champs Informix SQL

array **ifx_fieldtypes** (int result_id)

ifx_fieldtypes retourne un tableau associatif avec les noms des champs comme clés, et les types SQL comme valeurs. En cas d'erreur, retourne FALSE .

Nom de champs et type SQL.

```

<?php
$types = ifx_fieldtypes($resultid);

```

```

if (!isset($types)) {
    /* ... erreur ... */
}
foreach ($types as $fname => $val) {
    echo "$fname:\t type = $val\n";
}
?>

```

8.51.21 ifx_free_blob() : Supprime un objet BLOB

int **ifx_free_blob** (int bid)

ifx_free_blob supprime l'objet BLOB bid . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.51.22 ifx_free_char() : Supprime un objet char

int **ifx_free_char** (int bid)

ifx_free_char supprime l'objet char bid . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.51.23 ifx_free_result() : Libère les ressources prises par un résultat

int **ifx_free_result** (int result_id)

ifx_free_result libère les ressources prises par le résultat result_id . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.51.24 ifx_get_blob() : Retourne le contenu d'un objet BLOB

int **ifx_get_blob** (int bid)

ifx_get_blob retourne le contenu de l'objet BLOB associé à bid .

8.51.25 ifx_get_char() : Retourne le contenu d'un objet char

int **ifx_get_char** (int bid)

ifx_get_char retourne le contenu de l'objet associé à l'identifiant bid .

8.51.26 ifx_getsqlca() : Retourne le contenu de la variable sqlca.sqlerrd[0..5] après une requête

array **ifx_getsqlca** (int result_id)

ifx_getsqlca retourne une pseudo-ligne (tableau associatif) avec sqlca.sqlerrd[0] à sqlca.sqlerrd[5] après la requête associée result_id .

result_id est un identifiant valide de résultat retourné par `ifx_query` ou `ifx_prepare`.

Pour les requêtes INSERT, UPDATE et DELETE, les valeurs retournées sont celles fixées par le serveur après avoir exécuté la requête. Cela donne accès au nombre de lignes affectées, ainsi qu'au numéro de série d'insertion. Pour les requêtes de type SELECT, les valeurs retournées sont celles qui ont été préparées. Utiliser cette fonction économise l'exécution d'une requête "select dbinfo('sqlca.sqlerrdx')", étant donné qu'elle retourne les valeurs qui ont été sauveées par le pilote ifx au moment approprié.

Lire les valeurs de sqlca.sqlerrd[x]

```
<?php
/* On suppose que la première colonne d'une table
'quelconque' est un numéro de série */
$qid = ifx_query("insert into sometable values(0, '2nd column', 'another column' ", $connid);
if (!$qid) {
    // ... erreur ...
}
$sqlca = ifx_getsqlca($qid);
$serial_value = $sqlca["sqlerrdl"];
echo "Le numéro de série de la valeur insérée
      est : " . $serial_value . "<br />\n";
?>
```

8.51.27 ifx_htmltbl_result() : Lit toutes les lignes d'un tableau, et le met sous la forme d'un tableau HTML

int `ifx_htmltbl_result` (int result_id , string html_table_options)

`ifx_htmltbl_result` lit toutes les lignes d'un tableau et le met sous la forme d'un tableau HTML, ou renvoie FALSE en cas d'erreur.

Affiche les lignes avec des balises HTML. Le second argument permet de modifier les options de la balise <table>.

Affichage sous la forme d'une table HTML

```
<?php
$rid = ifx_prepare("select * from emp where name like " . $name,
                  $connid, IFX_SCROLL);
if (!$rid) {
    // ... erreur ...
}
$rowcount = ifx_affected_rows($rid);
if ($rowcount > 1000) {
    printf ("Trop de lignes dans le résultat : (%d)\n<br />", $rowcount);
    die ("Recommencez votre requête <br />\n");
}
if (!ifx_do($rid)) {
    // ... erreur ...
}
ifx_htmltbl_result($rid, 'border="2"');
ifx_free_result($rid);
?>
```

8.51.28 ifx_nullformat() : Modifie le mode par défaut de lecture des valeurs

void **ifx_nullformat** (int mode)

ifx_nullformat modifie le mode par défaut de lecture des valeurs. Le mode "0" retourne "", et le mode "1" retourne "NULL".

8.51.29 ifx_num_fields() : Retourne le nombre de colonnes dans une requête

int **ifx_num_fields** (int result_id)

ifx_num_fields retourne le nombre de colonnes dans la requête result_id ou FALSE en cas d'erreur.

Après avoir préparé ou exécuté une requête, cette fonction retourne le nombre de colonnes dans la requête.

8.51.30 ifx_num_rows() : Compte le nombre de lignes déjà lues dans un résultat

int **ifx_num_rows** (int result_id)

ifx_num_rows compte le nombre de lignes déjà lues dans le résultat result_id après ifx_query ou ifx_do .

8.51.31 ifx_pconnect() : Ouvre une connexion persistante à un serveur Informix

int **ifx_pconnect** (string database , string userid , string password)

ifx_pconnect retourne un identifiant positif de connexion persistante à un serveur Informix, ou FALSE , en cas d'erreur.

ifx_pconnect se comporte de manière très similaire à ifx_connect avec deux différences importantes.

ifx_pconnect se comporte exactement comme ifx_connect lorsque PHP n'est pas un module Apache. Lors de la connexion, la fonction va chercher une connexion déjà ouverte avec le même hôte, le même nom d'utilisateur et le même mot de passe. Si elle en trouve une, elle retournera un identifiant de cette connexion au lieu d'en ouvrir une nouvelle.

Deuxièmement, la connexion au serveur SQL ne sera pas automatiquement refermée à la fin de l'exécution du script. Au contraire, le lien va rester ouvert (ifx_close ne fermera pas les connexions établies avec ifx_pconnect).

Ainsi, ce type de lien est appelé 'persistant'.

Voir aussi ifx_connect .

8.51.32 `ifx_prepare()` : Prépare une requête SQL pour l'exécution

int `ifx_prepare` (string `query` , int `conn_id` , int `cursor_def` , mixed `blobidarray`)

`ifx_prepare` retourne un entier identifiant de résultat `result_id` à utiliser avec `ifx_do` . Modifie la valeur de `affected_rows` , pour accès ultérieur avec `ifx_affected_rows` .

`ifx_prepare` prépare la requête `query` sur la connexion `conn_id` . Pour les requêtes de type "select-type" un pointeur de résultat est déclaré et ouvert. L'option `cursor_type` permet de choisir le type de pointeur : "scroll" et/ou "hold". Les valeurs peuvent être combinées ensemble (IFX_SCROLL, IFX_HOLD).

Le nombre de lignes affectées (estimé ou exact) est enregistré, pour être lu avec la fonction `ifx_affected_rows` .

Si vous avez une colonne de type BLOB (BYTE ou TEXT) dans une requête de modification, vous pouvez passer un paramètre `blobidarray` qui contiendra les identifiants des BLOB à modifier, et vous devrez remplacer cette colonne par un point d'interrogation (?) dans la requête.

Si le contenu d'une colonne est de type TEXT (ou BYTE) vous pouvez aussi utiliser les fonctions `ifx_textasvarchar` et `ifx_byteasvarchar` . Cela vous permettra d'utiliser les colonnes TEXT (ou BYTE) comme des colonnes de type VARCHAR (mais plus long, tout de même), et vous n'aurez pas besoin de l'identifiant de BLOB.

Avec les fonctions `ifx_textasvarchar` et `ifx_byteasvarchar` (valeurs par défaut), les requêtes SELECT retourneront des identifiants de BLOB. Cet identifiant peut être une chaîne ou un fichier, suivant la configuration (voir plus loin).

Voir aussi `ifx_do` .

8.51.33 `ifx_query()` : Envoie une requête Informix

int `ifx_query` (string `query` , int `link_identifier` , int `cursor_type` , mixed `blobidarray`)

`ifx_query` retourne un identifiant positif de résultat Informix en cas de succès, et FALSE en cas d'erreur.

L'entier de type "identifiant de résultat" est utilisé par d'autres fonctions pour lire les résultats. Pour un exemple, reportez-vous à `ifx_affected_rows` afin de connaître le nombre de lignes affectées.

`ifx_query` envoie une requête au serveur actif courant, associé à l'identifiant de connexion `link_identifier` .

Exécute la requête `query` sur la connexion `conn_id` . Pour les requêtes de type SELECT, un pointeur est déclaré, et ouvert. L'option `cursor_type` permet de choisir le type de pointeur, "scroll" et/ou "hold". `cursor_type` accepte les deux valeurs séparées, et leur combinaison. Les requêtes d'autre type sont à exécution immédiate.

Le nombre de lignes affectées (estimé ou exact) est enregistré pour être lu avec `ifx_affected_rows` .

Si vous avez une colonne de type BLOB (BYTE ou TEXT) dans une requête de modification, vous pouvez passer un paramètre `blobidarray` qui contiendra les identifiants des BLOB à modifier, et vous devrez remplacer cette colonne par un point d'interrogation (?) dans la requête.

Si le contenu d'une colonne est de type TEXT (ou BYTE) vous pouvez aussi utiliser les fonctions `ifx_textasvarchar` et `ifx_byteasvarchar`. Cela vous permettra d'utiliser les colonnes TEXT (ou BYTE) comme des colonnes de type VARCHAR (mais plus long, tout de même), et vous n'aurez pas besoin de l'identifiant de BLOB.

Avec les fonctions `ifx_textasvarchar` et `ifx_byteasvarchar` (valeurs par défaut), les requêtes SELECT retourneront des identifiants de BLOB. Ces identifiants peuvent être une chaîne ou un fichier, suivant la configuration (voir plus loin).

Afficher toutes les lignes de la table "ordres" sous la forme html du serveur IFX

```
<?php
ifx_textasvarchar(1);      // Utilisation du mode "text mode" pour les BLOBs
$res_id = ifx_query("select * from orders", $conn_id);
if (! $res_id) {
    printf("Impossible de sélectionner des lignes dans : %s\n<br />%s<br />\n", ifx_error());
    ifx_errormsg();
    die;
}
ifx_htmltbl_result($res_id, "border=\"1\"");
ifx_free_result($res_id);
?>
```

Insertion de valeurs dans la table "catalogue" IFX

```
<?php

// créer un identifiant de BLOB pour une colonne de type BYTE et une de type TEXT
$textid = ifx_create_blob(0, 0, "Colonne Text en mémoire");
$byteid = ifx_create_blob(1, 0, "Colonne Byte en mémoire");

// Stocke l'identifiant du BLOB dans le tableau BLOBid
$BLOBidarray[] = $textid;
$BLOBidarray[] = $byteid;

// exécute la requête
$query = "insert into catalog (stock_num, manu_code, " .
        "cat_descr,cat_picture) values(1,'HRO',?,?)";
$res_id = ifx_query($query, $conn_id, $BLOBidarray);
if (! $res_id) {
    // ... erreur ...
}

// libération du résultat
ifx_free_result($res_id);
?>
```

Voir aussi [ifx_connect](#).

8.51.34 ifx_textasvarchar() : Choisit le mode par défaut des objets text

```
void ifx_textasvarchar ( int mode )
```

`ifx_textasvarchar` modifie le mode par défaut des objets TEXT. Le mode "0" retournera un identifiant de BLOB et le mode "1" retourne le BLOB sous la forme d'un (gros) varchar.

8.51.35 ifx_update_blob() : Modifie le contenu d'un objet BLOB

```
bool ifx_update_blob ( int bid , string content )
```

ifx_update_blob modifie le contenu de l'objet BLOB repéré par son identifiant bid . content est une chaîne contenant les nouvelles données. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.51.36 ifx_update_char() : Modifie le contenu d'un objet char

int **ifx_update_char** (int bid , string content)

ifx_update_char modifie le contenu de l'objet char repéré par son identifiant bid . content est une chaîne avec les nouvelles données. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.51.37 ifxus_close_slob() : Ferme un objet SLOB

int **ifxus_close_slob** (int bid)

ifxus_close_slob ferme l'objet SLOB représenté par son identifiant bid . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.51.38 ifxus_create_slob() : Crée un objet SLOB et l'ouvre

int **ifxus_create_slob** (int mode)

ifxus_create_slob crée un objet SLOB et l'ouvre. Les modes valides sont : 1 = LO_RDONLY, 2 = LO_WRONLY, 4 = LO_APPEND, 8 = LO_RDWR, 16 = LO_BUFFER, 32 = LO_NOBUFFER -> ou une combinaison des précédents. Vous pouvez aussi utiliser les constantes suivantes : IFX_LO_RDONLY, IFX_LO_WRONLY etc. ifxus_create_slob retourne FALSE en cas d'erreur et, sinon, l'identifiant de l'objet SLOB.

8.51.39 ifxus_free_slob() : Supprime un objet SLOB

int **ifxus_free_slob** (int bid)

ifxus_free_slob supprime un objet SLOB. bid est l'identifiant de l'objet SLOB. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.51.40 ifxus_open_slob() : Ouvre un objet SLOB

int **ifxus_open_slob** (int bid , int mode)

ifxus_open_slob ouvre un objet SLOB. bid est un identifiant d'objet SLOB. Les modes valides sont : 1 = LO_RDONLY, 2 = LO_WRONLY, 4 = LO_APPEND, 8 = LO_RDWR, 16 = LO_BUFFER, 32 = LO_NOBUFFER -> ou une combinaison des valeurs précédentes. ifxus_open_slob retourne l'identifiant du SLOB en cas de réussite, et FALSE sinon.

8.51.41 ifxus_read_slob() : Lit n bytes d'un objet SLOB

int **ifxus_read_slob** (int bid , int nbytes)

ifxus_read_slob lit nbytes octets de l'objet SLOB bid . bid est un identifiant d'objet SLOB existant, et nbytes est le nombre d'octets à lire. ifxus_read_slob retourne FALSE en cas d'erreur et, sinon, une chaîne de caractères.

8.51.42 ifxus_seek_slob() : Fixe le fichier courant ou la position courante

int **ifxus_seek_slob** (int bid , int mode , int offset)

ifxus_seek_slob modifie le fichier courant, ou la position du pointeur de fichier, pour un objet SLOB ouvert. bid est un identifiant d'objet SLOB. Les modes valides sont : 0 = LO_SEEK_SET, 1 = LO_SEEK_CUR, 2 = LO_SEEK_END et offset est un octet d'offset. ifxus_seek_slob retourne FALSE en cas d'erreur et, sinon, la position du pointeur de fichier.

8.51.43 ifxus_tell_slob() : Retourne le fichier courant ou la position courante

int **ifxus_tell_slob** (int bid)

ifxus_tell_slob retourne le fichier courant, ou la position courante d'un objet SLOB ouvert. bid est un identifiant d'objet SLOB. ifxus_tell_slob retourne FALSE en cas d'erreur et, sinon, la position du pointeur de fichier.

8.51.44 ifxus_write_slob() : Ecrit une chaîne dans un objet SLOB

int **ifxus_write_slob** (int bid , string content)

ifxus_write_slob écrit une chaîne dans un objet SLOB. bid est un identifiant d'objet SLOB et content sont les données à écrire. ifxus_write_slob retourne FALSE en cas d'erreur et, sinon, le nombre d'octets écrits.

8.52 Fonctions d'administration d'IIS

8.52.1 Introduction

Cette extension PECL n'est pas intégrée à PHP. Cette extension n'est disponible que pour Windows. Elle fournit des fonctions d'administration de Microsoft Internet Information Server (IIS). Cette extension inclut des fonctions pour créer des sites web et des dossiers virtuels, ainsi que pour configurer la sécurité et les extensions de scripts. Ces fonctions ont été ajoutées en PHP 4.

Pour utiliser ces fonctions, vous devez activer la bibliothèque `php_iisfunc.dll` dans le fichier `php.ini`. Vous pouvez télécharger ces bibliothèques DLL pour ces extensions PECL depuis la page de téléchargement de PHP sous sur la page <http://snaps.php.net/>.

8.52.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.52.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.52.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.52.5 Types de ressources

Cette extension ne définit aucune ressource.

8.52.6 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [iis_add_server](#)
- [iis_get_dir_security](#)
- [iis_get_script_map](#)
- [iis_get_server_by_comment](#)
- [iis_get_server_by_path](#)
- [iis_get_server_rights](#)
- [iis_get_service_state](#)

- [iis_remove_server](#)
- [iis_set_app_settings](#)
- [iis_set_dir_security](#)
- [iis_set_script_map](#)
- [iis_set_server_rights](#)
- [iis_start_server](#)
- [iis_start_service](#)
- [iis_stop_server](#)
- [iis_stop_service](#)

8.52.8 iis_get_dir_security() : Lit la configuration de sécurité du dossier

int [iis_get_dir_security](#) (int server_instance , string virtual_path)

8.52.9 iis_get_script_map() : Lit le mappage de script dans un dossier virtuel donné

string [iis_get_script_map](#) (int server_instance , string virtual_path , string script_extension)

8.52.10 iis_get_server_by_comment() : Retourne le numéro d'instance associé à un Comment

int [iis_get_server_by_comment](#) (string comment)

8.52.11 iis_get_server_by_path() : Retourne le numéro d'instance associé avec le Path

int [iis_get_server_by_path](#) (string path)

Chaque serveur virtuel sous IIS est représenté par un numéro d'instance. [iis_get_server_by_path](#) trouve ce numéro d'instance à partir du chemin physique jusqu'à la racine du dossier.

8.52.12 iis_get_server_rights() : Lit les droits du serveur

int [iis_get_server_rights](#) (int server_instance , string virtual_path)

8.52.13 iis_get_service_state() : Retourne l'état du service défini par ServiceId

int [iis_get_service_state](#) (string service_id)

8.52.14 iis_remove_server() : Supprime le serveur IIS représenté par ServerInstance

int **iis_remove_server** (int server_instance)

8.52.15 iis_set_app_settings() : Crée un espace d'exécution pour une application dans un dossier virtuel

int **iis_set_app_settings** (int server_instance , string virtual_path , string application_scope)

8.52.16 iis_set_dir_security() : Modifie la configuration de sécurité du dossier

int **iis_set_dir_security** (int server_instance , string virtual_path , int directory_flags)

8.52.17 iis_set_script_map() : Modifie le mappage de script pour un dossier virtuel

int **iis_set_script_map** (int server_instance , string virtual_path , string script_extension , string engine_path , int allow_scripting)

8.52.18 iis_set_server_rights() : Modifie les droits du serveur

int **iis_set_server_rights** (int server_instance , string virtual_path , int directory_flags)

8.52.19 iis_start_server() : Démarre le serveur web virtuel

int **iis_start_server** (int server_instance)

8.52.20 iis_start_service() : Démarre le service IIS identifié par ServiceId

int **iis_start_service** (string service_id)

8.52.21 iis_stop_server() : Stoppe le serveur web virtuel

int **iis_stop_server** (int server_instance)

8.52.22 iis_stop_service() : Stoppe le service IIS identifié par ServiceId

int **iis_stop_service** (string service_id)

8.53 Images

8.53.1 Introduction

PHP ne se limite pas à la génération de pages HTML. Il peut aussi servir à créer et manipuler des images, dans un grand choix de formats, comme GIF , PNG , JPEG , WBMP et XPM . Et PHP peut même générer directement des images pour le navigateur, avec la bibliothèque GD . GD et PHP auront aussi besoin d'autres bibliothèques, en fonction des formats que vous souhaitez utiliser.

Vous pouvez utiliser les fonctions PHP pour obtenir les tailles des images aux formats JPEG , GIF , PNG , SWF , TIFF et JPEG2000 .

Note

Lisez la section sur les pré-requis pour savoir comment améliorer les possibilités de lecture, modification et écriture d'images, ainsi que les possibilités de lecture d'images prises par des appareils numériques.

8.53.2 Pré-requis

Si vous disposez de la bibliothèque GD (disponible à <http://www.boutell.com/gd/>) vous pourrez aussi créer et manipuler des images.

Les formats des images que vous pourrez manipuler dépendent de la version de GD que vous installerez, et de toutes autres bibliothèques dont GD a besoin pour traiter ces images. Les versions antérieures à la version 1.6 supportent le GIF , mais pas le PNG . A partir des version de GD supérieures à 1.6 et inférieures à 2.0.28, c'est le contraire. Le support du GIF a été rajouté à partir de la verison 2.0.28.

Note

Depuis PHP 4.3, il existe une version de GD qui est distribuée avec PHP. Cette version contient des fonctionnalités supplémentaires, comme les canaux alpha, et il est recommandé de l'utiliser de préférence à la bibliothèque externe, car elle est mieux supportée, et bien plus stable.

Vous pouvez aussi améliorer GD en lui ajoutant des formats d'images supplémentaires.

Format d'image	Bibliothèque à télécharger	Notes
gif		Uniquement supporté en versions de GD antérieures à la version 1.6 et supérieures ou égales à la version 2.0.28. Le support des images GIF est disponible en lecture seule depuis PHP 4.3.0, et dans la version de la bibliothèque qui est fournie avec la distribution de PHP. L' écriture est supportée depuis PHP 4.3.9 et PHP 5.0.1.
jpeg-6b	ftp://ftp.uu.net/graphics/jpeg/	

png	http://www.libpng.org/pub/png/libpng.html	Uniquement supporté avec GD plus récente que la version 1.6.
xpm	ftp://metalab.unc.edu/pub/Linux/libs/X/INDEX.html	Il est probable que vous ayez déjà cette bibliothèque disponible si votre système dispose d'un environnement X.

Vous pouvez aussi améliorer GD en lui ajoutant des fonctionnalités de manipulation de polices. Les bibliothèques suivantes sont supportées :

Bibliothèque de police	Téléchargement	Notes
FreeType 1.x	http://www.freetype.org/	
FreeType 2	http://www.freetype.org/	
T1lib	ftp://sunsite.unc.edu/pub/Linux/libs/graphics/	Support des polices de Type 1.

Si vous avez compilé PHP avec l'option `--enable-exif` vous serez capable d'accéder aux données stockées dans les en-têtes des images JPEG et TIFF . De cette façon, vous pouvez lire les méta données générées par les appareils numériques. Ces fonctions ne requièrent par la bibliothèque GD .

8.53.3 Installation

Pour activer le support de GD, vous devez compiler PHP avec l'option `--with-gd[=DIR]` , où DIR est le dossier d'installation de GD. Il est recommandé d'utiliser la version de GD qui est distribuée avec PHP, en utilisant simplement l'option `--with-gd` . La bibliothèque GD requiert libpng et libjpeg pour compiler.

Sous Windows, vous devez inclure la bibliothèque `php_gd2.dll` comme extension dans le fichier `php.ini` . La bibliothèque `php_gd.dll` a été supprimée de PHP 4.3.2. Notez aussi que les fonctions conseillées `truecolor`, comme `imagecreatetruecolor` , requièrent GD2.

Note
Pour activer le support exif sur Windows, <code>php_mbstring.dll</code> doit être chargée avant <code>php_exif.dll</code> dans le <code>php.ini</code> .

Pour désactiver le support de GD en PHP 3, ajoutez simplement l'option `--without-gd` dans votre ligne de configuration.

Augmentez les possibilités des GD de gérer d'autres formats d'images en spécifiant les options `--with-XXXX` de compilation suivantes :

Format d'image	Option de compilation
jpeg-6b	Pour activer le support de la bibliothèque JPEG-6b, ajouter l'option <code>--with-jpeg-dir=DIR</code> .
png	Pour activer le support de la bibliothèque PNG, ajouter l'option <code>--with-png-dir=DIR</code> . Notez que libpng requiert la bibliothèque <code>zlib</code> et, donc, il vous faudra ajouter aussi <code>--with-zlib-dir[=DIR]</code> dans votre ligne de compilation.
xpm	Pour activer le support de la bibliothèque XPM, ajoutez l'option <code>--with-xpm-dir=DIR</code> . Si le script de compilation n'est pas capable de trouver les bibliothèques nécessaires, il vous faudra ajouter le chemin vers les bibliothèques X11.

Note

Lorsque vous compilez PHP avec libpng, vous devez utiliser la même version que celle liée à la bibliothèque GD.

Augmentez les possibilités de GD pour qu'elle manipule différents types de polices de caractères en ajoutant les options --with-XXXX de compilation suivantes :

Bibliothèque	Option de configuration
FreeType 1.x	Pour activer le support de FreeType 1.x, ajoutez l'option --with-ttf[=DIR] .
FreeType 2	Pour activer le support de FreeType 2, ajoutez l'option --with-freetype-dir=DIR .
T1lib	Pour activer le support de T1lib (Type 1 fonts), ajoutez l'option --with-t1lib[=DIR] .
Chaînes TrueType	Pour activer le support des chaînes de caractères TrueType, ajoutez l'option --enable-gd-native-ttf .

8.53.4 Configuration à l'exécution

Il n'y a pas de configuration spécifique pour les images mais vous êtes peut être intéressé par les directives de l' [extension exif](#) .

8.53.5 Types de ressources

Cette extension ne définit aucune ressource.

8.53.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

IMG_GIF ([entier](#))
 IMG_JPG ([entier](#))
 IMG_JPEG ([entier](#))
 IMG_PNG ([entier](#))
 IMG_WBMP ([entier](#))
 IMG_XPM ([entier](#))
 IMG_COLOR_TILED ([entier](#))
 IMG_COLOR_STYLED ([entier](#))
 IMG_COLOR_BRUSHED ([entier](#))
 IMG_COLOR_STYLEDBRUSHED ([entier](#))
 IMG_COLOR_TRANSPARENT ([entier](#))
 IMG_ARC_ROUNDED ([entier](#))
 IMG_ARC_PIE ([entier](#))
 IMG_ARC_CHORD ([entier](#))
 IMG_ARC_NOFILL ([entier](#))
 IMG_ARC_EDGED ([entier](#))
 IMAGETYPE_GIF ([entier](#))
 IMAGETYPE_JPEG ([entier](#))
 IMAGETYPE_PNG ([entier](#))
 IMAGETYPE_SWF ([entier](#))
 IMAGETYPE_PSD ([entier](#))
 IMAGETYPE_BMP ([entier](#))
 IMAGETYPE_WBMP ([entier](#))

IMAGETYPE_XBM ([entier](#))
 IMAGETYPE_TIFF_II ([entier](#))
 IMAGETYPE_TIFF_MM ([entier](#))
 IMAGETYPE_IFF ([entier](#))
 IMAGETYPE_JB2 ([entier](#))
 IMAGETYPE_JPC ([entier](#))
 IMAGETYPE_JP2 ([entier](#))
 IMAGETYPE_JPX ([entier](#))
 IMAGETYPE_SWC ([entier](#))

8.53.7 Exemples

Création d'une image PNG avec PHP

```

<?php

header("Content-type: image/png");
$string = $_GET['text'];
$im     = imagecreatefrompng("images/button1.png");
$orange = imagecolorallocate($im, 220, 210, 60);
$px     = (imagesx($im) - 7.5 * strlen($string)) / 2;
imagestring($im, 3, $px, 9, $string, $orange);
imagepng($im);
imagedestroy($im);

?>

```

Cet exemple doit être appelé depuis une page HTML avec une balise image telle que : `` . Le script ci-dessus, `button.php` , prend la chaîne "texte" et l'inscrit sur le fond d'image appelé "images/button1.png" , puis l'affiche. C'est une méthode très pratique pour éviter de redessiner un nouveau bouton, dès que vous changez son texte. De cette façon, ils sont générés dynamiquement.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [gd_info](#)
- [getimagesize](#)
- [image_type_to_extension](#)
- [image_type_to_mime_type](#)
- [image2wbmp](#)
- [imagealphablending](#)
- [imageantialias](#)
- [imagearc](#)
- [imagechar](#)
- [imagecharup](#)
- [imagecolorallocate](#)
- [imagecolorallocatealpha](#)
- [imagecolorat](#)
- [imagecolorclosest](#)
- [imagecolorclosestalpha](#)
- [imagecolorclosesthw](#)

- [imagecolordeallocate](#)
- [imagecolorexact](#)
- [imagecolorexactalpha](#)
- [imagecolormatch](#)
- [imagecolorresolve](#)
- [imagecolorresolvealpha](#)
- [imagecolorset](#)
- [imagecolorsforindex](#)
- [imagecolorstotal](#)
- [imagecolortransparent](#)
- [imageconvolution](#)
- [imagecopy](#)
- [imagecopymerge](#)
- [imagecopymergegray](#)
- [imagecopyresampled](#)
- [imagecopyresized](#)
- [imagecreate](#)
- [imagecreatefromgd](#)
- [imagecreatefromgd2](#)
- [imagecreatefromgd2part](#)
- [imagecreatefromgif](#)
- [imagecreatefromjpeg](#)
- [imagecreatefrompng](#)
- [imagecreatefromstring](#)
- [imagecreatefromwbmp](#)
- [imagecreatefromxbm](#)
- [imagecreatefromxpm](#)
- [imagecreatetruecolor](#)
- [imagedashedline](#)
- [imagedestroy](#)
- [imageellipse](#)
- [imagefill](#)
- [imagefilledarc](#)
- [imagefilledellipse](#)
- [imagefilledpolygon](#)
- [imagefilledrectangle](#)
- [imagefilltoborder](#)
- [imagefilter](#)
- [imagefontheight](#)
- [imagefontwidth](#)
- [imageftbbox](#)
- [imagefttext](#)
- [imagegammacorrect](#)
- [imagegd](#)
- [imagegd2](#)
- [imagegif](#)
- [imageinterlace](#)
- [imageistruecolor](#)
- [imagejpeg](#)
- [imagelayereffect](#)
- [imageline](#)
- [imageloadfont](#)
- [imagepalettecopy](#)
- [imagepng](#)

- [imagepolygon](#)
- [imagepsbbox](#)
- [imagepscropyfont](#)
- [imagepsencodefont](#)
- [imagepsextendfont](#)
- [imagepsfreefont](#)
- [imagepsloadfont](#)
- [imagepslantfont](#)
- [imagepstext](#)
- [imagerectangle](#)
- [imagerotate](#)
- [imagesavealpha](#)
- [imagesetbrush](#)
- [imagesetpixel](#)
- [imagesetstyle](#)
- [imagesetthickness](#)
- [imagesettile](#)
- [imagestring](#)
- [imagestringup](#)
- [imagesx](#)
- [imagesy](#)
- [imagetruecolortopalette](#)
- [imaggittfbbox](#)
- [imaggittfttext](#)
- [imaggitypes](#)
- [imagewbmp](#)
- [imagexbm](#)
- [iptcembed](#)
- [iptcparse](#)
- [jpeg2wbmp](#)
- [png2wbmp](#)

8.53.9 [getimagesize\(\)](#) : Retourne la taille d'une image

array [getimagesize](#) (string filename , array imageinfo)

[getimagesize](#) va déterminer la taille des images de type GIF , JPG , PNG , SWF , SWC , PSD , TIFF , BMP , IFF , JP2 , JPX , JB2 , JPC , XBM , or WBMP et en retourner les dimensions, le type d'image et une chaîne type height/width à placer dans une balise HTML IMG normale.

Si l'accès à filename est impossible ou bien si filename n'est pas une image valide, [getimagesize](#) retournera FALSE et générera une erreur de niveau E_WARNING .

Note
Le support de JPC , JP2 , JPX , JB2 , XBM et WBMP a été ajouté en PHP 4.3.2. Le support de SWC , lui, en PHP 4.3.0 et le support de TIFF en PHP 4.2.0.
Note
Le support de JPEG 2000 a été ajouté en PHP 4.3.2. Notez que JPC et JP2 sont capables d'avoir des composants avec une profondeur de bit différente. Dans ce cas, la valeur de "bits" est la plus grande profondeur de bit rencontrée. De même, les fichiers JP2 disposent du support de multiple JPEG 2000 codestreams . Dans ce cas, getimagesize retourne les valeurs pour le premier codestream rencontré à la racine du fichier.

Note

La fonction `getimagesize` ne requiert pas la bibliothèque GD.

`getimagesize` retourne un tableau de 4 éléments. L'index 0 contient la largeur. L'index 1 contient la longueur. L'index 2 contient le type de l'image : 1 = GIF , 2 = JPG , 3 = PNG , 4 = SWF , 5 = PSD , 6 = BMP , 7 = TIFF (Ordre des octets Intel), 8 = TIFF (Ordre des octets Motorola), 9 = JPC , 10 = JP2 , 11 = JPX , 12 = JB2 , 13 = SWC , 14 = IFF . Ces valeurs correspondent aux constantes `IMAGETYPE` qui ont été ajoutées en PHP 4.3. L'index 3 contient la chaîne à placer dans les balises `IMG` : `height="xxx" width="yyy"` .

Exemple avec `getimagesize`

```
<?php
list($width, $height, $type, $attr) = getimagesize("img/flag.jpg");
echo "<img src=\"img/flag.jpg\" alt=\"Exemple de getimagesize()\" $attr />";
?>
```

Le support des URL a été ajouté en PHP 4.0.5.

`getimagesize` avec une URL

```
<?php
$size = getimagesize("http://www.example.com/gifs/logo.gif");
// Si le nom du fichier comporte des espaces, encodez-le !
$size = getimagesize("http://www.example.com/gifs/lo%20go.gif");
?>
```

Avec les images JPG , deux en-têtes supplémentaires sont retournés : `channels` et `bits` . `channels` vaudra 3 avec les images RGB, et 4 avec les images CMYK. `bits` est le nombre de bits pour chaque couleur.

Depuis PHP 4.3, `bits` et `channels` sont présents pour tous types d'images. Cependant, la présence de ces valeurs peut mener à la confusion. Par exemple, une image GIF utilise toujours trois canaux par pixel, mais le nombre de bits par pixel ne peut être calculé dans le cas d'une image animée GIF ayant une table de couleur globale.

Certains formats peuvent ne contenir aucune image, ou bien plusieurs. Dans ces cas-là, `getimagesize` peut ne pas être capable de déterminer correctement la taille de l'image. `getimagesize` retourne alors zéro comme taille de hauteur et largeur.

Depuis PHP 4.3, `getimagesize` retourne aussi un paramètre additionnel, `mime` , qui correspond au type MIME d'une image. Cette information peut être utilisée pour envoyer l'en-tête HTTP correct.

`getimagesize` et types MIME

```
<?php
$size = getimagesize($filename);
$fp = fopen($filename, "rb");
if ($size && $fp) {
    header("Content-type: {$size['mime']}");
    fpassthru($fp);
    exit;
} else {
    // error
}
?>
```

Le paramètre optionnel `imageinfo` permet d'extraire des informations supplémentaires du fichier image. Actuellement, cette option va retourner différents marqueurs JPG APP dans un tableau associatif. Certains programmes utilisent ces marqueur APP pour préciser les informations dans les balises HTML. Un marqueur commun est le marqueur APP13, décrit à <http://www.iptc.org/> . Vous

pouvez utiliser la fonction [iptcparse](#) pour analyser ce marqueur, et obtenir des informations intelligibles.

```

getimagesize qui retourne IPTC
<?php
$size = getimagesize("testimg.jpg", $info);
if (isset($info["APP13"])) {
    $iptc = iptcparse($info["APP13"]);
    var_dump($iptc);
}
?>
    
```

Voir aussi [image_type_to_mime_type](#) , [exif_imagetype](#) , [exif_read_data](#) et [exif_thumbnail](#) .

8.53.10 **image_type_to_extension()** : Retourne l'extension du fichier pour le type d'image

string **image_type_to_extension** (int imagetype , bool include_dot)

Attention
 Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.53.11 **image_type_to_mime_type()** : Lit le Mime-Type d'un type d'image

string **image_type_to_mime_type** (int imagetype)

[image_type_to_mime_type](#) détermine le type Mime (Mime-Type), à utiliser dans l'en-tête HTTP Content-type . `imagetype` est une constante, représentant un type d'image et retournée par [getimagesize](#) , [exif_read_data](#) , [exif_thumbnail](#) ou [exif_imagetype](#) .

```

image_type_to_mime_type (fichier)
<?php
header("Content-type: " . image_type_to_mime_type(IMAGETYPE_PNG));
?>
    
```

Les valeurs de retour sont les suivantes :

imagetype	Valeur retournée
IMAGETYPE_GIF	image/gif
IMAGETYPE_JPEG	image/jpeg
IMAGETYPE_PNG	image/png
IMAGETYPE_SWF	application/x-shockwave-flash
IMAGETYPE_PSD	image/psd
IMAGETYPE_BMP	image/bmp
IMAGETYPE_TIFF_IL (ordre intel)	image/tiff
IMAGETYPE_TIFF_MM (ordre motorola)	image/tiff
IMAGETYPE_JPC	application/octet-stream
IMAGETYPE_JP2	image/jp2
IMAGETYPE_JPX	application/octet-stream
IMAGETYPE_JB2	application/octet-stream
IMAGETYPE_SWC	application/x-shockwave-flash

IMAGETYPE_IFF	image/iff
IMAGETYPE_WBMP	image/vnd.wap.wbmp
IMAGETYPE_XBM	image/xbm

Note

Cette fonction ne requiert pas la bibliothèque GD.

Voir aussi [getimagesize](#) , [exif_imagetype](#) , [exif_read_data](#) et [exif_thumbnail](#) .

8.53.12 image2wbmp() : Crée une image WBMP

int **image2wbmp** (resource image , string filename , int threshold)

[image2wbmp](#) crée une image WBMP à partir de l'image image . Si le paramètre filename est fourni, elle sera créée dans ce fichier, sinon, elle sera envoyée au navigateur. image est une image valide, créée avec une fonction de création d'images, comme la fonction [imagecreatetruecolor](#) .

Le nom de fichier filename est optionnel et, s'il est omis, l'image sera renvoyée directement au navigateur. En retournant un en-tête HTTP Content-Type : IMAGE/VND.WAP.WBMP avec la fonction [header](#) , vous pouvez créer des images WBMP avec vos scripts PHP.

Exemple avec [image2wbmp](#)

```
<?php
$file = 'php.png';
$image = imagecreatefrompng($file);

header('Content-type: ' . image_type_to_mime_type(IMAGETYPE_WBMP));
image2wbmp($image); // Affiche le contenu directement

?>
```

Note

Le support WBMP n'est disponible qu'avec GD-1.8 ou plus récent.

Voir aussi [imagewbmp](#) .

8.53.13 imagealphablending() : Modifie le mode de blending d'une image

bool **imagealphablending** (resource image , bool blendmode)

[imagealphablending](#) fournit deux modes de dessin des images en vraies couleurs (truecolors). En mode "blending", le canal alpha de chaque couleur est fournie à chaque fonction de dessin, tel que [imagesetpixel](#) peut déterminer sa transparence. GD va alors automatiquement mixer la couleur à ce point, et stocker le résultat dans l'image. Le pixel résultant est alors opaque. En mode non-mixant, la couleur est copiée littéralement avec ses informations de canal alpha, et remplace le pixel de destination. Le mixage n'est pas disponible avec les images à palette. Si blendmode vaut TRUE , alors le mode de mixage sera activé, sinon il sera désactivé.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.

8.53.14 imageantialias() : Utiliser ou non les fonctions d'antialias

bool **imageantialias** (resource im , bool on)

Active les méthodes de schéma rapide d'antialias et de polygones câblés. Il ne supporte pas les composants alpha. Il fonctionne en utilisant une opération directe de mélange. Il ne fonctionne qu'avec les images truecolor.

L'épaisseur et le style ne sont pas supportés.

L'utilisation des primitives antialias avec des arrière-plans transparents peut se terminer avec des résultats imprévus. La méthode de mélange utilise la couleur d'arrière-plan comme tout autre couleur. Les faiblesses du support du composant alpha font qu'il n'est pas autorisé de méthode d'antialias basée sur l'alpha.

Note

Cette fonction n'est disponible que si PHP est compilé avec la version embarquée de la bibliothèque GD.

Voir aussi [imagecreatetruecolor](#) .

8.53.15 imagearc() : Dessine une ellipse partielle

bool **imagearc** (resource image , int cx , int cy , int w , int h , int s , int e , int color)

[imagearc](#) dessine une ellipse partielle, centrée sur cx , cy (le coin en haut à gauche est l'origine (0,0)) dans l'image référencée par image . w et h spécifient la largeur et la hauteur de l'ellipse, tandis que le début et la fin de l'arc sont donnés en degrés, par les arguments s et e . color est la couleur utilisée pour dessiner l'ellipse.

Dessin d'un cercle avec [imagearc](#)

```
<?php
// Création d'une image 200*200
$img = imagecreatetruecolor(200, 200);

// Allocation de couleurs
$white = imagecolorallocate($img, 255, 255, 255);
$black = imagecolorallocate($img, 0, 0, 0);

// Dessin d'un cercle noir
imagearc($img, 100, 100, 150, 150, 0, 360, $black);

// Affichage au navigateur
header("Content-type: image/png");
imagepng($img);

// Libération de la mémoire
imagedestroy($img);
?>
```

Voir aussi [imageellipse](#) , [imagefilledellipse](#) et [imagefilledarc](#) .

8.53.16 imagechar() : Dessine un caractère horizontalement

bool **imagechar** (resource image , int font , int x , int y , string c , int color)

imagechar dessine le premier caractère de la chaîne c dans l'image id avec le coin supérieur gauche placé à la position x , y (le coin en haut à gauche est l'origine (0,0)) avec la couleur color . Si la police est 1, 2, 3, 4 ou 5, une police intégrée sera utilisée (plus le chiffre est grand, plus la police est grande).

Exemple avec imagechar

```
<?php
$im = imagecreatetruecolor(100, 100);
$string = 'PHP';
$bg = imagecolorallocate($im, 255, 255, 255);
$black = imagecolorallocate($im, 0, 0, 0);

// affiche un "P" noir dans le coin gauche en haut
imagechar($im, 1, 0, 0, $string, $black);

header('Content-type: image/png');
imagepng($im);
?>
```

Voir aussi imagecharup et imagedloadfont .

8.53.17 imagecharup() : Dessine un caractère verticalement

bool **imagecharup** (resource image , int font , int x , int y , string c , int color)

imagecharup dessine le premier caractère de la chaîne c verticalement dans l'image id avec le coin supérieur gauche placé à la position (x , y) (le coin en haut à gauche est l'origine (0,0)), avec la couleur color . Si la police est 1, 2, 3, 4 ou 5, une police intégrée sera utilisée (plus le chiffre est grand, plus la police est grande).

Exemple avec imagecharup

```
<?php
$im = imagecreatetruecolor(100, 100);
$string = 'Notez que la première lettre est un N';
$bg = imagecolorallocate($im, 255, 255, 255);
$black = imagecolorallocate($im, 0, 0, 0);

// affiche un "Z" noir sur un fond blanc
imagecharup($im, 3, 10, 10, $string, $black);

header('Content-type: image/png');
imagepng($im);
?>
```

Voir aussi imagechar et imagedloadfont .

8.53.18 imagecolorallocate() : Alloue une couleur pour une image

int **imagecolorallocate** (resource image , int red , int green , int blue)

imagecolorallocate retourne un identifiant de couleur, représentant la couleur composée avec les couleurs RGB (red , green , blue). L'argument image est le résultat de la fonction imagecreatetruecolor . Les paramètres red , green et blue sont les valeurs respectives des composants rouge, vert et bleue de la couleur voulue. Ces paramètres sont des entiers compris entre 0 et 255 ou des hexadécimaux compris entre 0x00 and 0xFF. imagecolorallocate doit être appelée pour créer chaque couleur qui sera représentée par image .

Note

Le premier appel à imagecolorallocate remplit la couleur de fond.

Exemple avec imagecolorallocate

```

<?php
$im = imagecreatetruecolor('example.jpg');
// Le fond de l'image est en rouge
$background = imagecolorallocate($im, 255, 0, 0);

// On définit des couleurs avec des entiers ..
$white = imagecolorallocate($im, 255, 255, 255);
$black = imagecolorallocate($im, 0, 0, 0);

// .. ou des hexadécimaux
$white = imagecolorallocate($im, 0xFF, 0xFF, 0xFF);
$black = imagecolorallocate($im, 0x00, 0x00, 0x00);

?>

```

imagecolorallocate retourne -1 si l'allocation échoue.

Voir aussi imagecolorallocatealpha et imagecolordeallocate .

8.53.19 imagecolorallocatealpha() : Alloue une couleur à une image

int **imagecolorallocatealpha** (resource image , int red , int green , int blue , int alpha)

imagecolorallocatealpha se comporte comme imagecolorallocate avec en plus le paramètre de transparence alpha qui prend une valeur entre 0 et 127 . 0 indique une opacité complète tandis que 127 indique une transparence complète.

imagecolorallocatealpha retourne FALSE si l'allocation a échoué.

Exemple avec imagecolorallocatealpha

```

<?php
$size = 300;
$image=imagecreatetruecolor($size, $size);

// quelque chose pour obtenir un fond blanc avec une bordure noire
$back = imagecolorallocate($image, 255, 255, 255);
$border = imagecolorallocate($image, 0, 0, 0);
imagefilledrectangle($image, 0, 0, $size - 1, $size - 1, $back);
imagerectangle($image, 0, 0, $size - 1, $size - 1, $border);

$yellow_x = 100;

```

```

$yellow_y = 75;
$red_x    = 120;
$red_y    = 165;
$blue_x   = 187;
$blue_y   = 125;
$radius   = 150;

// alloue des couleurs avec des valeurs alpha
$yellow = imagecolorallocatealpha($image, 255, 255, 0, 75);
$red    = imagecolorallocatealpha($image, 255, 0, 0, 75);
$blue   = imagecolorallocatealpha($image, 0, 0, 255, 75);

// Dessine 3 ellipses
imagefilledellipse($image, $yellow_x, $yellow_y, $radius, $radius, $yellow);
imagefilledellipse($image, $red_x, $red_y, $radius, $radius, $red);
imagefilledellipse($image, $blue_x, $blue_y, $radius, $radius, $blue);

// Ne pas oublier d'envoyer un header correct
header('Content-type: image/png');

// et finalement, afficher le résultat
imagepng($image);
imagedestroy($image);
?>

```

Note

Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.

Voir aussi [imagecolorallocate](#) et [imagecolordeallocate](#) .

8.53.20 imagecolorat() : Retourne l'index de la couleur d'un pixel donné

int **imagecolorat** (resource image , int x , int y)

[imagecolorat](#) retourne l'index de la couleur du pixel situé aux coordonnées (x , y), dans l'image image .

Si PHP supporte la bibliothèque GD version 2.0 ou plus récent, et que l'image est une image en TrueColor, cette fonction retourne la valeur RGB du pixel, sous forme d'un entier. Utiliser les opérateurs de bits et les masques pour distinguer le rouge, du vert et du bleu :

Accès aux valeurs RGB

```

<?php
$im = ImageCreateFromPng("rockym.png");
$rgb = ImageColorAt($im, 100, 100);
$r = ($rgb >> 16) & 0xFF;
$g = ($rgb >> 8) & 0xFF;
$b = $rgb & 0xFF;
?>

```

Voir aussi [imagecolorset](#) et [imagecolorsforindex](#) .

8.53.21 imagecolorclosest() : Retourne l'index de la couleur la plus proche d'une couleur donnée

int **imagecolorclosest** (resource image , int red , int green , int blue)

imagecolorclosest retourne l'index de la couleur de la palette qui est la plus proche de la valeur RGB passée.

La "distance" entre la couleur souhaitée et les couleurs de la palette est calculée en considérant l'espace RGB comme un espace à 3 dimensions.

Si vous créez l'image depuis un fichier, seules les couleurs utilisées dans l'image seront résolues. Les couleurs uniquement présentes dans la palette ne seront pas résolues.

Voir aussi imagecolorexact .

8.53.22 imagecolorclosestalpha() : Retourne la couleur la plus proche, en tenant compte du canal alpha

int **imagecolorclosestalpha** (resource image , int red , int green , int blue , int alpha)

imagecolorclosestalpha retourne l'index de la couleur, dans la palette de l'image image , la plus proche de la couleur spécifiée par les autres paramètres, au format RGB et de canal alpha alpha .

Note

Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.

Voir aussi imagecolorexactalpha .

8.53.23 imagecolorclosesthwb() : Lit l'index de la couleur la plus proche d'une couleur spécifiée avec sa teinte, blanc et noir

int **imagecolorclosesthwb** (resource image , int red , int green , int blue)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.53.24 imagecolordeallocate() : Supprime une couleur d'une image

bool **imagecolordeallocate** (resource image , int color)

imagecolordeallocate désalloue la couleur color précédemment allouée avec la fonction imagecolorallocate , pour l'image image .

Exemple avec <u>imagecolordeallocate</u>
--

<pre><?php \$white = imagecolorallocate(\$im, 255, 255, 255); imagecolordeallocate(\$im, \$white); ?></pre>

Voir aussi imagecolorallocate et imagecolorallocatealpha .

8.53.25 imagecolorexact() : Retourne l'index de la couleur donnée

int **imagecolorexact** (resource image , int red , int green , int blue)

8.53.21 imagecolorclosest() : Retourne l'index de la couleur la plusproche d'une couleur donnée848

[imagecolorexact](#) retourne l'index de la couleur spécifiée dans la palette de l'image image .

Si la couleur n'existe pas dans cette palette, [imagecolorexact](#) retourne -1.

Si vous créez l'image depuis un fichier, seules les couleurs utilisées dans l'image seront résolues. Les couleurs uniquement présentes dans la palette ne seront pas résolues.

Voir aussi [imagecolorclosest](#) .

8.53.26 imagecolorexactalpha() : Retourne l'index d'une couleur avec son canal alpha

int **imagecolorexactalpha** (resource image , int red , int green , int blue , int alpha)

[imagecolorexactalpha](#) retourne l'index de la couleur fournie au format RGB et son canal alpha alpha , dans l'image image .

Si la couleur n'existe pas dans la palette de l'image, [imagecolorexactalpha](#) retourne -1.

Note
Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.
Voir aussi imagecolorclosestalpha .

8.53.27 imagecolormatch() : Fait correspondre un peu plus les couleurs de la version palette d'une image aux couleurs de sa version truecolor

bool **imagecolormatch** (resource image1 , resource image2)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.
image1 doit être Truecolor, image2 doit être Palette, et image1 et image2 doivent avoir la même taille.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note
Cette fonction n'est disponible que si PHP est compilé avec la version embarquée de la bibliothèque GD.
Note
Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.
Voir aussi imagecreatetruecolor .

8.53.28 imagecolorresolve() : Retourne l'index de la couleur donnée, ou la plus proche possible

int **imagecolorresolve** (resource image , int red , int green , int blue)

[imagecolorresolve](#) retourne un index de couleur à tous les coups. Soit il arrive à trouver la couleur demandée dans la palette, soit il recherche la couleur la plus proche.

Si vous créez l'image depuis un fichier, seules les couleurs utilisées dans l'image seront résolues. Les couleurs uniquement présentes dans la palette ne seront pas résolues.

Voir aussi [imagecolorclosest](#) .

8.53.29 imagecolorresolvealpha() : Retourne un index de couleur ou son alternative la plus proche, y compris le canal alpha

int **imagecolorresolvealpha** (resource image , int red , int green , int blue , int alpha)

[imagecolorresolvealpha](#) retourne toujours un index de couleur, disponible dans la palette de l'image image : soit c'est la couleur exacte, soit c'est la meilleure approximation.

Note

Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.

Voir aussi [imagecolorclosestalpha](#) .

8.53.30 imagecolorset() : Change la couleur dans une palette à l'index donné

void **imagecolorset** (resource image , int index , int red , int green , int blue)

[imagecolorset](#) permet d'attribuer à un index d'une palette une couleur spécifique. C'est une fonction très pratique pour effectuer du remplissage de couleur sans le faire réellement.

Voir aussi [imagecolorat](#) .

8.53.31 imagecolorsforindex() : Retourne la couleur associée à un index

array **imagecolorsforindex** (resource image , int index)

[imagecolorsforindex](#) retourne un tableau associatif avec les couleurs rouge (red) , vert (green), bleu (blue) et alpha qui contiennent les valeurs des couleurs correspondantes.

Exemple avec imagecolorsforindex
--

```
<?php
// on ouvre une image
$im = imagecreatefrompng('nexen.png');

// on obtient une couleur
$start_x = 40;
$start_y = 50;
$color_index = imagecolorat($im, $start_x, $start_y);

// on la rend humainement lisible
$color_tran = imagecolorsforindex($im, $color_index);

// quelle est-elle ?
```

8.53.28 imagecolorresolve() : Retourne l'index de la couleur donnée, ou la plus proche possible 850

```
print_r($color_tran);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
  [red] => 226
  [green] => 222
  [blue] => 252
  [alpha] => 0
)
```

Voir aussi [imagecolorat](#) et [imagecolorexact](#) .

8.53.32 imagecolorstotal() : Calcule le nombre de couleurs d'une palette

int **imagecolorstotal** (resource image)

[imagecolorstotal](#) retourne le nombre de couleurs de la palette pour l'image image .

Voir aussi [imagecolorat](#) et [imagecolorsforindex](#) .

8.53.33 imagecolortransparent() : Définit la couleur transparente

int **imagecolortransparent** (resource image , int color)

[imagecolortransparent](#) permet de choisir la couleur transparente d'une image, et de lui donner la valeur de color . image est un identifiant d'image, retourné par [imagecreatetruecolor](#) et color est un identifiant de couleur retourné par [imagecolorallocate](#) .

Note

La couleur de transparence est une propriété de l'image, elle n'est pas une propriété de la couleur. Une fois que vous avez défini la couleur de transparence, chaque région de l'image de cette couleur que vous avez dessiné précédemment sera transparente.

L'identifiant de la nouvelle (ou courante) couleur transparente est retourné.

Note

La transparence est copiée uniquement avec la fonction [imagecopymerge](#) et les images en couleur vraies, non pas avec la fonction [imagecopy](#) ou les images de palette.

8.53.34 imageconvolution() : Applique une matrice de la convolution 3x3, en utilisant la division de coefficient et l'excentrage

bool **imageconvolution** (resource image , array matrix3x3 , float div , float offset)

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Note

Cette fonction n'est disponible que si PHP est compilé avec la version embarquée de la bibliothèque GD.

8.53.35 imagecopy() : Copie une partie d'une image

bool **imagecopy** (resource dst_im , resource src_im , int dst_x , int dst_y , int src_x , int src_y , int src_w , int src_h)

Copie une partie de l'image src_im sur l'image de destination dst_im , en commençant aux coordonnées src_x , src_y et sur la largeur de src_w et la hauteur de src_h . La portion ainsi définie sera copiée et placée aux coordonnées dst_x et dst_y .

8.53.36 imagecopymerge() : Copie et fusionne une partie d'une image

bool **imagecopymerge** (resource dst_im , resource src_im , int dst_x , int dst_y , int src_x , int src_y , int src_w , int src_h , int pct)

imagecopymerge copie une partie de l'image src_im dans l'image de destination dst_im en commençant aux coordonnées (src_x , src_y), avec la largeur src_w et la hauteur src_h . La zone de l'image ainsi définie sera copiée aux coordonnées (dst_x , dst_y), dans l'image de destination. Les deux images seront fusionnées suivant le paramètre pct , qui peut valoir de 0 à 100. Si pct = 0, aucune action n'est faite, alors que si pct = 100, imagecopymerge se comporte exactement comme imagecopy pour les images de palette, tandis qu'il implémente la transparence alpha pour les images en couleur vraies.

Note

<u>imagecopymerge</u> a été ajoutée en PHP 4.0.6.

8.53.37 imagecopymergegray() : Copie et fusionne une partie d'une image en niveaux de gris

bool **imagecopymergegray** (resource dst_im , resource src_im , int dst_x , int dst_y , int src_x , int src_y , int src_w , int src_h , int pct)

imagecopymergegray copie une partie de l'image src_im dans l'image de destination dst_im commençant aux coordonnées (src_x , src_y), avec la largeur src_w et la hauteur src_h . La zone de l'image ainsi définie sera copiée aux coordonnées (dst_x , dst_y), dans l'image de destination. Les deux images seront fusionnées suivant le paramètre pct , qui peut valoir de 0 à 100. Si pct = 0, aucune action n'est faite, alors que si pct = 100, imagecopymerge se comporte exactement comme imagecopy .

imagecopymergegray est identique à la fonction imagecopymerge , hormis le fait que lors de la fusion, le "hue" de l'image sera conservé grâce à la conversion de la zone dans l'image de destination en gris, avant l'opération de copie.

Note

<u>imagecopymergegray</u> a été ajoutée en PHP 4.0.6.

8.53.38 `imagecopyresampled()` : Copie, redimensionne, rééchantillonne une image

bool **imagecopyresampled** (resource dst_image , resource src_image , int dst_x , int dst_y , int src_x , int src_y , int dst_w , int dst_h , int src_w , int src_h)

`imagecopyresampled` copie une zone rectangulaire de l'image `src_im` vers l'image `dst_im` . Durant la copie, la zone est rééchantillonnée de manière à conserver la clarté de l'image durant une réduction. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

`dst_im` est l'image de destination, `src_im` est l'image source. Si les hauteurs et largeurs des source et destination diffèrent, l'image copiée sera étirée de manière appropriée. Les coordonnées sont celles du coin supérieur gauche. `imagecopyresampled` peut servir à copier des zones d'une image vers elle-même, (si `dst_image` est la même que `src_image`) mais si les régions se chevauchent, les résultats sont imprévisibles.

Note

Il y a un problème dû aux limitations de la taille de la palette (255 + 1 couleurs différentes). Filtrer ou rééchantillonner une image demande plus de 255 couleurs, une approximation est alors utilisée pour calculer le nouveau nombre de couleurs. Avec une palette, si une nouvelle couleur ne peut être allouée, la couleur la plus proche (en théorie) est utilisée. Ce n'est pas toujours la couleur la plus proche visuellement. Cela peut générer des problèmes étranges, comme des images blanches. Pour éviter ce problème, passez en image TrueColor, comme celles générée par la fonction <code>imagecreatetruecolor</code> .

Note

Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.

8.53.39 `imagecopyresized()` : Copie et redimensionne une partie d'une image

bool **imagecopyresized** (resource dst_image , resource src_image , int dst_x , int dst_y , int src_x , int src_y , int dst_w , int dst_h , int src_w , int src_h)

`imagecopyresized` copie une partie rectangulaire d'une image dans une autre image de destination. `dst_image` est l'image de destination, `src_image` est l'image source. Si les dimensions de la source et de la destination ne sont pas égales, un étirement adéquat est effectué pour faire correspondre les deux. Les coordonnées fournies sont définies par rapport au coin supérieur gauche. Cette fonction peut être utilisée pour recopier des régions à l'intérieur d'une même image (si `dst_image` et `src_image` sont identiques), mais si les régions se chevauchent, le résultat risque d'être incohérent.

Note

Il y a un problème dû aux limitations de la taille de la palette (255 + 1 couleurs différentes). Filtrer ou rééchantillonner une image demande plus de 255 couleurs, une approximation est alors utilisée pour calculer le nouveau nombre de couleurs. Avec une palette, si une nouvelle couleur ne peut être allouée, la couleur la plus proche (en théorie) est utilisée ; ce n'est pas toujours celle qui est la plus proche visuellement. Cela peut générer des problèmes étranges, comme des images blanches. Pour éviter ce problème, passez en image TrueColor, comme celles générée par la fonction <code>imagecreatetruecolor</code> .

8.53.40 imagecreate() : Crée une nouvelle image à palette

resource **imagecreate** (int x_size , int y_size)

imagecreate retourne un identifiant d'image représentant une image vide, de largeur x_size et longueur y_size .

Nous recommandons l'utilisation de imagecreatetruecolor à la place de imagecreate .

Création d'une image GD et affichage de cette image

```
<?php
header ("Content-type: image/png");
$im = @imagecreate (100, 50)
    or die ("Impossible d'initialiser la bibliothèque GD");
$background_color = imagecolorallocate ($im, 255, 255, 255);
$text_color = imagecolorallocate ($im, 233, 14, 91);
imagestring ($im, 1, 5, 5, "A Simple Text String", $text_color);
imagepng ($im);
imagedestroy($im);
?>
```

Voir aussi imagedestroy et imagecreatetruecolor .

8.53.41 imagecreatefromgd() : Crée une nouvelle image à partir d'un fichier GD ou d'une URL

resource **imagecreatefromgd** (string filename)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Array

Attention

Les versions Windows de PHP antérieures à la version 4.3.0 ne supportent pas l'accès aux fichiers distants avec cette fonction, même si <u>allow_url_fopen</u> est activé.
--

8.53.42 imagecreatefromgd2() : Crée une nouvelle image à partir d'un fichier GD2 ou d'une URL

resource **imagecreatefromgd2** (string filename)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Note

Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.

Array

Attention

Les versions Windows de PHP antérieures à la version 4.3.0 ne supportent pas l'accès aux fichiers distants avec cette fonction, même si <u>allow_url_fopen</u> est activé.
--

8.53.43 imagecreatefromgd2part() : Crée une nouvelle image à partir d'une partie de fichier GD2 ou d'une URL

resource **imagecreatefromgd2part** (string filename , int srcX , int srcY , int width , int height)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.
Note
Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.
Array
Attention
Les versions Windows de PHP antérieures à la version 4.3.0 ne supportent pas l'accès aux fichiers distants avec cette fonction, même si <code>allow_url_fopen</code> est activé.

8.53.44 imagecreatefromgif() : Crée une nouvelle image à partir d'un fichier ou d'une URL

resource **imagecreatefromgif** (string filename)

`imagecreatefromgif` retourne un identifiant d'image qui représente l'image obtenue à partir du fichier dont le nom est donné par filename .

`imagecreatefromgif` retourne une chaîne vide en cas d'échec. Il va aussi retourner une erreur qui va afficher un lien brisé dans un navigateur. Pour simplifier le débogage, utilisez le code suivant, qui retourne une erreur GIF :

Exemple de gestion des erreurs durant la création d'image
<pre><?php function LoadGif (\$imgname) { \$im = @imagecreatefromgif (\$imgname); /* Tentative d'ouverture */ if (!\$im) { /* Test d'échec */ \$im = imagecreatetruecolor (150, 30); /* Création d'une image vide */ \$bgc = imagecolorallocate (\$im, 255, 255, 255); \$tc = imagecolorallocate (\$im, 0, 0, 0); imagefilledrectangle (\$im, 0, 0, 150, 30, \$bgc); /* Affichage d'un message d'erreur */ imagestring (\$im, 1, 5, 5, "Erreur au chargement de l'image \$imgname", \$tc); } return \$im; } ?></pre>
Note
Le support du GIF a été supprimé de la bibliothèque GD dans sa version 1.6 et a été rajouté depuis la version 2.0.28. Cette fonction n'est donc pas disponible entre ces deux versions.
Array
Attention
Les versions Windows de PHP antérieures à la version 4.3.0 ne supportent pas l'accès aux fichiers distants avec cette fonction, même si <code>allow_url_fopen</code> est activé.

8.53.45 imagecreatefromjpeg() : Crée une nouvelle image JPEG à partir d'un fichier ou d'une URL

resource **imagecreatefromjpeg** (string filename)

imagecreatefromjpeg retourne un identifiant d'image représentant une image obtenue à partir du fichier filename .

imagecreatefromjpeg retourne une chaîne vide en cas d'échec. Elle affiche aussi un message d'erreur, qui est représenté comme un lien brisé dans un navigateur web. Pour faciliter le débogage, voici une erreur JPEG :

Exemple de gestion d'erreur lors de la création d'image

```
<?php
function loadjpeg($imgname) {
    $im = @imagecreatefromjpeg($imgname); /* Tentative d'ouverture */
    if (!$im) { /* Vérification */
        $im = imagecreatetruecolor(150, 30); /* Création d'une image blanche */
        $bgc = imagecolorallocate($im, 255, 255, 255);
        $tc = imagecolorallocate($im, 0, 0, 0);
        imagefilledrectangle($im, 0, 0, 150, 30, $bgc);
    // Affichage d'un message d'erreur
        imagestring($im, 1, 5, 5, "Erreur de chargement de l'image $imgname", $tc);
    }
    return $im;
}
?>
```

Note

Le support JPEG n'est disponible que si PHP a été compilé avec GD-1.8 ou supérieur.

Array

Attention

Les versions Windows de PHP antérieures à la version 4.3.0 ne supportent pas l'accès aux fichiers distants avec cette fonction, même si [allow_url_fopen](#) est activé.

8.53.46 imagecreatefrompng() : Crée une nouvelle image PNG à partir d'un fichier ou d'une URL

resource **imagecreatefrompng** (string filename)

imagecreatefrompng retourne un identifiant d'image représentant une image obtenue à partir du fichier filename .

imagecreatefrompng retourne une chaîne vide en cas d'échec. Elle affiche aussi un message d'erreur, qui s'affiche comme un lien brisé dans un navigateur web. Pour faciliter le débogage, voici une erreur PNG :

Exemple de gestion d'erreur lors de la création d'image

```
<?php
function LoadPNG($imgname) {
    $im = @imagecreatefrompng($imgname); /* Tentative d'ouverture */
    if (!$im) { /* Vérification */
        $im = imagecreatetruecolor(150, 30); /* Création d'une image blanche */
        $bgc = imagecolorallocate($im, 255, 255, 255);
        $tc = imagecolorallocate($im, 0, 0, 0);
        imagefilledrectangle($im, 0, 0, 150, 30, $bgc);
    }
}
```

```

        /* Affichage d'un message d'erreur */
        imagestring($im, 1, 5, 5, "Erreur de chargement de l'image $imgname", $tc);
    }
    return $im;
}
?>

```

Array**Attention**

Les versions Windows de PHP antérieures à la version 4.3.0 ne supportent pas l'accès aux fichiers distants avec cette fonction, même si `allow_url_fopen` est activé.

8.53.47 imagecreatefromstring() : Crée une image à partir d'une chaîne

resource **imagecreatefromstring** (string image)

`imagecreatefromstring` retourne un identifiant d'image représentant l'image obtenu depuis la chaîne image . Le type de l'image sera automatiquement détecté si vous avez compilé PHP avec les supports : JPEG, PNG, GIF, WBMP et GD2.

8.53.48 imagecreatefromwbmp() : Crée une nouvelle image WBMP à partir d'un fichier ou d'une URL

resource **imagecreatefromwbmp** (string filename)

`imagecreatefromwbmp` retourne une ressource d'image PHP, représentant l'image filename .

`imagecreatefromwbmp` retourne une chaîne vide en cas d'erreur. Il retourne aussi un message d'erreur qui s'affiche comme un lien mort dans un navigateur. Pour aider au débogage, l'exemple suivant va produire une erreur WBMP :

Exemple de gestion des erreurs durant la création d'une image

```

<?php
function loadwbmp($imgname) {
    $im = @imagecreatefromwbmp($imgname); /* Tentative d'ouverture */
    if (!$im) { /* Vérification que cela s'est bien passé */
        $im = imagecreatetruecolor(20, 20); /* Crée une image blanche */
        $bgc = imagecolorallocate($im, 255, 255, 255);
        $tc = imagecolorallocate($im, 0, 0, 0);
        imagefilledrectangle($im, 0, 0, 10, 10, $bgc);
        // Affiche le message d'erreur
        imagestring($im, 1, 5, 5, "Erreur de chargement de $imgname", $tc);
    }
    return $im;
}
?>

```

Note

Le support WBMP n'est disponible que si PHP a été compilé avec GD-1.8 ou supérieur.

Array**Attention**

Les versions Windows de PHP antérieures à la version 4.3.0 ne supportent pas l'accès aux fichiers distants avec cette fonction, même si `allow_url_fopen` est activé.

8.53.49 imagecreatefromxbm() : Crée une nouvelle image XBM à partir d'un fichier ou d'une URL

resource **imagecreatefromxbm** (string filename)

imagecreatefromxbm retourne un identifiant d'image représentant l'image obtenue à partir du fichier filename .

Array

Attention

Les versions Windows de PHP antérieures à la version 4.3.0 ne supportent pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé.

8.53.50 imagecreatefromxpm() : Crée une nouvelle image XPM à partir d'un fichier ou d'une URL

resource **imagecreatefromxpm** (string filename)

imagecreatefromxpm retourne un identifiant d'image représentant l'image obtenue à partir du fichier filename .

Note

Cette fonction n'est disponible que si PHP est compilé avec la version embarquée de la bibliothèque GD.

Array

Attention

Les versions Windows de PHP antérieures à la version 4.3.0 ne supportent pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé.

8.53.51 imagecreatetruecolor() : Crée une nouvelle image en vraies couleurs

resource **imagecreatetruecolor** (int x_size , int y_size)

imagecreatetruecolor retourne une ressource représentant une image noire de largeur x_size , et de hauteur y_size .

Suivant votre version de PHP et de GD, cette fonction est définie ou non. De PHP version 4.0.6 à la version 4.1.x, cette fonction existe toujours si le module GD est chargé, mais l'appeler sans avoir GD2 d'installé fera que PHP lancera une erreur fatale et sortira. Avec PHP 4.2.x, ce comportement est différent et PHP lancera une alerte au lieu d'une erreur. Pour les autres versions de PHP, cette fonction n'est définie que si une version correcte de GD est installée.

Création d'un flux d'image GD, et affichage

```
<?php
header ("Content-type: image/png");
$im = @imagecreatetruecolor (50, 100)
    or die ("Impossible de créer un flux d'image GD");
$text_color = imagecolorallocate ($im, 233, 14, 91);
imagestring ($im, 1, 5, 5, "Une simple chaîne de texte", $text_color);
imagepng ($im);
imagedestroy ($im);
```

8.53.49 imagecreatefromxbm() : Crée une nouvelle image XBM à partir d'un fichier ou d'une URL

?>

Note

Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.

Note

imagecreatetruecolor ne fonctionne pas avec le format GIF .

Voir aussi imagedestroy et imagecreate .

8.53.52 imagedashedline() : Dessine une ligne pointillée

bool **imagedashedline** (resource image , int x1 , int y1 , int x2 , int y2 , int color)

imagedashedline est obsolète. Utilisez plutôt une combinaison des fonctions imagesetstyle et imageline à la place.

8.53.53 imagedestroy() : détruit une image

bool **imagedestroy** (resource image)

imagedestroy libère toute la mémoire associée à l'image image . image est un identifiant d'image valide retourné par une fonction de création d'images comme la fonction imagecreatetruecolor .

8.53.54 imageellipse() : Dessine une ellipse

bool **imageellipse** (resource image , int cx , int cy , int w , int h , int color)

imageellipse dessine une ellipse centrée sur le point (cx , cy). Le coin supérieur gauche est aux coordonnées (0,0). L'image de dessin est image . w et h spécifient respectivement la largeur et la hauteur de l'ellipse. La couleur de dessin de l'ellipse est color .

Exemple avec imageellipse

```
<?php
// nouvelle image
$image = imagecreatetruecolor(400,300);

// remplissage du fond
$bg = imagecolorallocate($image,0,0,0);

// couleur de l'ellipse
$col_ellipse = imagecolorallocate($image,255,255,255);

// on dessine l'ellipse
imageellipse($image, 200, 150, 300, 200, $col_ellipse);

// on affiche l'image
header("Content-type: image/png");
imagepng($image);

?>
```

Note

`imageellipse` a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.2 ou plus récent qui peut être récupéré ici : <http://www.boutell.com/gd/> .

Voir aussi `imagefilledellipse` et `imagearc` .

8.53.55 imagefill() : Remplit

bool `imagefill` (resource image , int x , int y , int color)

`imagefill` effectue un remplissage avec la couleur color , dans l'image image , à partir du point de coordonnées (x , y) (le coin supérieur gauche est l'origine (0,0)).

8.53.56 imagefilledarc() : Dessine une ellipse partielle et la remplit

bool `imagefilledarc` (resource image , int cx , int cy , int w , int h , int s , int e , int color , int style)

`imagefilledarc` dessine une ellipse partielle, centrée sur le point (cx , cy). Le coin supérieur gauche est (0, 0), dans l'image image . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. w et h spécifient respectivement la largeur et la hauteur de l'ellipse, tandis que les points de début et de fin sont représentés par s et e , en degrés. L'argument style est un champ de bits, combiné avec l'opérateur OR :

1. IMG_ARC_PIE
2. IMG_ARC_CHORD
3. IMG_ARC_NOFILL
4. IMG_ARC_EDGED

IMG_ARC_PIE et IMG_ARC_CHORD sont mutuellement exclusives; IMG_ARC_CHORD ne fait que connecter les angles de début et de fin avec une ligne droite, tandis que IMG_ARC_PIE produit une ligne courbe. IMG_ARC_NOFILL indique que l'arc (ou corde) doit être dessiné mais pas rempli. IMG_ARC_EDGED , utilisé conjointement avec IMG_ARC_NOFILL , indique que les angles de début et de fin doivent être connectés au centre. Cette fonction est recommandée pour faire les graphiques de type camembert.

Création d'un camembert en 3D

```
<?php
// Création de l'image
$image = imagecreatetruecolor(100, 100);

// Allocation de quelques couleurs
$white = imagecolorallocate($image, 0xFF, 0xFF, 0xFF);
$gray = imagecolorallocate($image, 0xC0, 0xC0, 0xC0);
$darkgray = imagecolorallocate($image, 0x90, 0x90, 0x90);
$navy = imagecolorallocate($image, 0x00, 0x00, 0x80);
$darknavy = imagecolorallocate($image, 0x00, 0x00, 0x50);
$red = imagecolorallocate($image, 0xFF, 0x00, 0x00);
$darkred = imagecolorallocate($image, 0x90, 0x00, 0x00);

// Création de l'effet 3D
for ($i = 60; $i > 50; $i--) {
    imagefilledarc($image, 50, $i, 100, 50, 0, 45, $darknavy, IMG_ARC_PIE);
    imagefilledarc($image, 50, $i, 100, 50, 45, 75, $darkgray, IMG_ARC_PIE);
    imagefilledarc($image, 50, $i, 100, 50, 75, 360, $darkred, IMG_ARC_PIE);
}
```

```

imagefilledarc($image, 50, 50, 100, 50, 0, 45, $navy, IMG_ARC_PIE);
imagefilledarc($image, 50, 50, 100, 50, 45, 75, $gray, IMG_ARC_PIE);
imagefilledarc($image, 50, 50, 100, 50, 75, 360, $red, IMG_ARC_PIE);

// Affichage de l'image
header('Content-type: image/png');
imagepng($image);
imagedestroy($image);
?>

```

Note

Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.

8.53.57 imagefilledellipse() : Dessine une ellipse pleine

bool **imagefilledellipse** (resource image , int cx , int cy , int w , int h , int color)

imagefilledellipse dessine une ellipse centrée sur le point (cx , cy). Le coin supérieur gauche est aux coordonnées (0,0). L'image de dessin est image . w et h spécifient respectivement la largeur et la hauteur de l'ellipse. La couleur de remplissage de l'ellipse est color .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec imagefilledellipse

```

<?php

// nouvelle image
$image = imagecreatetruecolor(400,300);

// couleur de fond
$bg = imagecolorallocate($image,0,0,0);

// couleur de remplissage de l'ellipse
$col_ellipse = imagecolorallocate($image,255,255,255);

// on dessine l'ellipse blanche
imagefilledellipse($image, 200, 150, 300, 200, $col_ellipse);

// on affiche l'image
header("Content-type: image/png");
imagepng($image);

?>

```

Note

Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.

8.53.58 imagefilledpolygon() : Dessine un polygone rempli

bool **imagefilledpolygon** (resource image , array points , int num_points , int color)

imagefilledpolygon dessine un polygone rempli dans l'image image .

points est un tableau qui contient les coordonnées x et y du sommet des polygones.

Le paramètre num_points est le nombre total de sommets, et doit être supérieur à 3.

Exemple avec `imagefilledpolygon`

```

<?php
// définition du tableau de points pour le polygone
$values = array(
    40, 50, // Point 1 (x, y)
    20, 240, // Point 2 (x, y)
    60, 60, // Point 3 (x, y)
    240, 20, // Point 4 (x, y)
    50, 40, // Point 5 (x, y)
    10, 10 // Point 6 (x, y)
);

// Création d'une image
$image = imagecreatetruecolor(250, 250);

// quelques couleurs
$bg = imagecolorallocate($image, 200, 200, 200);
$blue = imagecolorallocate($image, 0, 0, 255);

// dessine le polygone
imagefilledpolygon($image, $values, 6, $blue);

// Affichage de l'image
header('Content-type: image/png');
imagepng($image);
imagedestroy($image);
?>

```

8.53.59 `imagefilledrectangle()` : Dessine un rectangle rempli

bool **imagefilledrectangle** (resource image , int x1 , int y1 , int x2 , int y2 , int color)

`imagefilledrectangle` dessine un rectangle de couleur color dans l'image image , en commençant par le sommet supérieur gauche (x1 , y1) et finissant au sommet inférieur droit (x2 , y2). Le coin supérieur gauche est l'origine (0, 0).

8.53.60 `imagefilltoborder()` : Remplit une région avec une couleur spécifique

bool **imagefilltoborder** (resource image , int x , int y , int border , int color)

`imagefilltoborder` remplit avec la couleur color toute la région à l'intérieur de la région limitée par la couleur border . Le point de départ est (x , y) (le coin supérieur gauche est l'origine (0,0)) et la couleur de la région est color .

8.53.61 `imagefilter()` : Applique un filtre à une image

bool **imagefilter** (resource src_im , int filtertype , int arg1 , int arg2 , int arg3)

`imagefilter` applique le filtre filtertype à l'image en utilisant les paramètres args1 , args2 et args3 lorsque cela est nécessaire.

Le paramètre filtertype peut prendre l'une des valeurs suivantes :

- `IMG_FILTER_NEGATE` : renverse toutes les couleurs de l'image.

- `IMG_FILTER_GRAYSCALE` : convertit l'image en grayscale .
- `IMG_FILTER_BRIGHTNESS` : modifie la luminosité de l'image. Utilisez le paramètre `args1` pour définir la luminosité.
- `IMG_FILTER_CONTRAST` : modifie le contraste de l'image. Utilisez le paramètre `args1` pour définir le contraste.
- `IMG_FILTER_COLORIZE` : Identique au paramètre `IMG_FILTER_GRAYSCALE` excepté que vous pouvez spécifier une couleur. Utilisez trois arguments séparés dans les paramètres `args1` , `args2` et `args3` sous la forme `red` , `blue` , `green` . L'intervalle pour chaque couleur est 0 - 255.
- `IMG_FILTER_EDGEDETECT` : utilise la détection des bords pour les mettre en évidence dans l'image.
- `IMG_FILTER_EMBOSS` : grave l'image en relief.
- `IMG_FILTER_GAUSSIAN_BLUR` : brouille l'image en utilisant la méthode gaussienne.
- `IMG_FILTER_SELECTIVE_BLUR` : brouille l'image.
- `IMG_FILTER_MEAN_REMOVAL` : son utilisation signifie le déplacement pour réaliser un effet "peu précis".
- `IMG_FILTER_SMOOTH` : rend l'image lisse (`smooth`). Utilisez le paramètre `args1` pour définir le degré de lisse.

Note

Cette fonction n'est disponible que si PHP est compilé avec la version embarquée de la bibliothèque GD.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec `imagefilter` en utilisant le paramètre `IMG_FILTER_GRAYSCALE`

```
<?php
$im = imagecreatefrompng('dave.png');
if ($im && imagefilter($im, IMG_FILTER_GRAYSCALE)) {
    echo 'Image convertie en grayscale.';
    imagepng($im, 'dave.png');
} else {
    echo 'La conversion en grayscale a échoué.';
}

imagedestroy($im);
?>
```

Exemple avec `imagefilter` en utilisant le paramètre `IMG_FILTER_BRIGHTNESS`

```
<?php
$im = imagecreatefrompng('sean.png');
if ($im && imagefilter($im, IMG_FILTER_BRIGHTNESS, 20)) {
    echo 'La luminosité de l\'image a été modifiée.';
    imagepng($im, 'sean.png');
} else {
    echo 'Echec lors de la modification de la luminosité.';
}

imagedestroy($im);
?>
```

Exemple avec `imagefilter` en utilisant le paramètre `IMG_FILTER_COLORIZE`

```
<?php
$im = imagecreatefrompng('philip.png');

/* R, G, B, donc 0, 255, 0 correspond au vert */
if ($im && imagefilter($im, IMG_FILTER_COLORIZE, 0, 255, 0)) {
```

```

    echo 'L\'image a été ombragée en vert avec succès.';
    imagepng($im, 'philip.png');
} else {
    echo 'Echec lors de la modification de l\'ombrage.';
}

imagedestroy($im);
?>

```

8.53.62 imagefontheight() : Retourne la hauteur de la police

int **imagefontheight** (int font)

imagefontheight retourne la hauteur de la police font en pixels.

Voir aussi imagefontwidth et imageloadfont .

8.53.63 imagefontwidth() : Retourne la largeur de la police

int **imagefontwidth** (int font)

imagefontwidth retourne la largeur de la police font en pixels.

Voir aussi imagefontheight et imageloadfont .

8.53.64 imageftbbox() : Calcule le rectangle d'encadrement pour un texte, en utilisant la police courante et freetype2

array **imageftbbox** (float size , float angle , string font_file , string text , array extrainfo)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.
Note
Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.
Note
Le paramètre extrainfo est optionnel depuis PHP 4.3.5.

8.53.65 imagefttext() : Ecrit du texte dans une image avec la police courante FreeType 2

array **imagefttext** (resource image , float size , float angle , int x , int y , int col , string font_file , string text , array extrainfo)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.
Note
Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.
Note
Le paramètre extrainfo est optionnel depuis PHP 4.3.5.

8.53.66 imagegammacorrect() : Applique une correction gamma à l'image

bool **imagegammacorrect** (resource image , float inputgamma , float outputgamma)

imagegammacorrect applique une correction gamma à l'image GD image . Le facteur d'entrée est inputgamma , et le facteur de sortie outputgamma .

8.53.67 imagegd() : Génère une image au format GD, vers le navigateur ou un fichier

bool **imagegd** (resource image , string filename)

imagegd génère une image GD vers le fichier filename . L'argument image est le retour de la fonction imagecreatetruecolor .

Le paramètre filename est optionnel et s'il est omis, le flux de l'image sera affiché directement.

Note
Le format GD est communément utilisé pour autoriser le chargement rapide des parties d'un image. Notez que le format GD est uniquement utilisable dans les applications compatibles GD. Voir aussi <u>imagegd2</u> .

8.53.68 imagegd2() : Génère une image au format GD2, vers le navigateur ou un fichier

bool **imagegd2** (resource image , string filename , int chunk_size , int type)

imagegd2 génère une image au format GD2, vers le fichier filename . Le paramètre image est le retour de la fonction imagecreatetruecolor .

Le paramètre optionnel type peut valoir IMG_GD2_RAW ou IMG_GD2_COMPRESSED . La valeur par défaut est IMG_GD2_RAW .

Note
Les paramètres optionnels chunk_size et type ont été ajoutés en PHP 4.3.2.
Note
Le format GD2 est communément utilisé pour charger rapidement les parties d'une image. Notez que le format GD2 est uniquement utilisable dans les applications compatibles GD2.
Note
Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.

Voir aussi imagegd .

8.53.69 imagegif() : Envoie une image GIF vers un navigateur ou un fichier

bool **imagegif** (resource image , string filename)

`imagegif` crée un fichier image GIF avec le nom filename d'après l'image image . L'argument image est un identifiant valide retourné par la fonction `imagecreatetruecolor` .

Le format de l'image sera GIF87A , à moins que l'image n'ait une couleur transparente (mise en place grâce à la fonction `imagecolortransparent`), ce qui fera qu'elle sera au format GIF89A .

Le format de l'image sera GIF87A tant que l'image n'est pas modifiée avec la fonction `imagecolortransparent` . Dans le cas d'une telle modification, le format de l'image sera GIF89A .

Le nom du fichier est optionnel et, dans ce cas, l'image sera transmise directement à la sortie standard. En envoyant un en-tête de type image/gifcontent-type , (grâce à la fonction `header`), vous pouvez créer des images avec des scripts PHP.

Note

Etant donné que toutes les fonctions GIF ont été supprimées de la bibliothèque GD version 1.6, cette fonction ne sera pas accessible si vous avez cette version de la bibliothèque. Le support du format GIF sera de retour milieu 2004 dans la bibliothèque GD. Pour plus d'informations, voir le site du [projet GD](#) .

Le code suivant vous permet d'écrire des scripts PHP plus portables : le type de GD est automatiquement détecté. Il remplace la séquence `Header("Content-type: image/gif"); ImageGif($im);` par un code plus souple :

Détection automatique des formats d'image supportés

```
<?php
  if (function_exists("imagegif")) {
    header("Content-type: image/gif");
    imagegif($im);
  }
  elseif (function_exists("imagejpeg")) {
    header("Content-type: image/jpeg");
    imagejpeg($im, "", 0.5);
  }
  elseif (function_exists("imagepng")) {
    header("Content-type: image/png");
    imagepng($im);
  } elseif (function_exists("imagewbmp")) {
    header("Content-type: image/vnd.wap.wbmp");
    imagewbmp($im);
  } else {
    die("Pas de support graphique avec PHP sur ce serveur");
  }
?>
```

Note

Depuis les versions 3.0.18 et 4.0.2, vous pouvez utiliser la fonction `imagetypes` au lieu de `function_exists` pour vérifier la présence des différents formats d'images supportés.

Détection automatique des formats d'image supportés avec `imagetypes`

```
if (imagetypes() & IMG_GIF) {
  header ("Content-type: image/gif");
  imagegif ($im);
}
elseif (imagetypes() & IMG_JPG) {
  ... etc.
```

Voir aussi `imagepng` , `imagewbmp` , `imagejpeg` et `imagetypes` .

8.53.70 `imageinterlace()` : Active ou désactive l'entrelacement

int `imageinterlace` (resource image , int *interlace*)

`imageinterlace` active ou désactive le bit d'entrelacement.

Si l'entrelacement est à 1, l'image image sera entrelacée, sinon, elle ne le sera pas. Si le format d'affichage de l'image est JPEG , l'image créée sera un JPEG progressif.

`imageinterlace` retourne l'état courant d'entrelacement de l'image.

8.53.71 `imageistruecolor()` : Détermine si une image est une image truecolor

bool `imageistruecolor` (resource image)

`imageistruecolor` détermine si l'image image est une image truecolor.

Note
Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure. Voir aussi <code>imagecreatetruecolor</code> .

8.53.72 `imagejpeg()` : Envoie une image JPEG vers un navigateur ou un fichier

bool `imagejpeg` (resource image , string *filename* , int *quality*)

`imagejpeg` envoie l'image GD (image) au format JPEG sur la sortie standard (typiquement, le navigateur web), ou si filename est fourni, l'envoi dans un fichier. image a été créé par `imagecreatetruecolor` .

Le nom du fichier est optionnel et, s'il n'est pas fourni, l'image sera transmise directement à la sortie standard. En envoyant un content-type image/jpeg (grâce à la fonction `header`), vous pouvez créer des images JPEG avec des scripts PHP.

Note
Le support JPEG n'est disponible que si PHP a été compilé avec GD-1.8 ou supérieur. quality est optionnel, et prend des valeurs entières de 0 (pire qualité, petit fichier) et 100 (meilleure qualité, gros fichier). Par défaut, la valeur est celle de la qualité IJG (75).

Si vous voulez générer des images JPEG progressives, vous devez activer l'entrelacement à l'aide de la fonction `imageinterlace` .

Voir aussi `imagepng` , `imagewbmp` , `imageinterlace` , `imagegif` et `imagetypes` .

8.53.73 `imagelayereffect()` : Active l'option d'alpha blending, pour utiliser les effets de libgd

bool `imagelayereffect` (resource image , int effect)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Note

Cette fonction n'est disponible que si PHP est compilé avec la version embarquée de la bibliothèque GD.

Note

Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.

8.53.74 imageline() : Dessine une ligne

bool **imageline** (resource image , int x1 , int y1 , int x2 , int y2 , int color)

imageline dessine une ligne depuis le point (x1 , y1) jusqu'au point (x2 , y2) (le coin supérieur gauche est l'origine (0,0)) dans l'image image et avec la couleur color .

Tracer une ligne fine

```
<?php
function imagelinethick($image, $x1, $y1, $x2, $y2, $color, $thick = 1)
{
    /* de cette manière, ca ne marche bien que pour les lignes orthogonales
    imagesetthickness($image, $thick);
    return imageline($image, $x1, $y1, $x2, $y2, $color);
    */
    if ($thick == 1) {
        return imageline($image, $x1, $y1, $x2, $y2, $color);
    }
    $t = $thick / 2 - 0.5;
    if ($x1 == $x2 || $y1 == $y2) {
        return imagefilledrectangle($image, round(min($x1, $x2) - $t), round(min($y1, $y2) - $t),
    }
    $k = ($y2 - $y1) / ($x2 - $x1); //y = kx + q
    $a = $t / sqrt(1 + pow($k, 2));
    $points = array(
        round($x1 - (1+$k)*$a), round($y1 + (1-$k)*$a),
        round($x1 - (1-$k)*$a), round($y1 - (1+$k)*$a),
        round($x2 + (1+$k)*$a), round($y2 - (1-$k)*$a),
        round($x2 + (1-$k)*$a), round($y2 + (1+$k)*$a),
    );
    imagefilledpolygon($image, $points, 4, $color);
    return imagepolygon($image, $points, 4, $color);
}
?>
```

Voir aussi [imagecreatetruecolor](#) et [imagecolorallocate](#) .

8.53.75 imageloadfont() : Charge une nouvelle police

int **imageloadfont** (string file)

imageloadfont charge une nouvelle police utilisateur et retourne un identifiant sur cette police. Cet identifiant sera toujours supérieur à 5, pour éviter les conflits avec les polices standard PHP. Retourne FALSE en cas d'erreur.

Le format des polices dépend actuellement du système d'exploitation. Ce qui signifie qu'il vous faut

générer des fichiers de polices pour la machine qui fait tourner PHP.

Position	Type de données C	Description
Octets 0-3	int	Nombre de caractères de la police
Octets 4-7	int	Valeur du premier caractère de la police (souvent 32 pour espace)
Octets 8-11	int	Largeur en pixels des caractères
Octets 12-15	int	Hauteur en pixels des caractères
Octets 16-	char	Tableau avec les données des caractères, un octet par pixel pour chaque caractère, avec un total de (nombre de caractères * largeur * hauteur) octets.

Exemple avec `imageloadfont`

```
<?php
$im = imagecreatetruecolor(50, 20);
$black = imagecolorallocate($im, 0, 0, 0);
$white = imagecolorallocate($im, 255, 255, 255);
imagefilledrectangle($im, 0, 0, 49, 19, $white);
$font = imageloadfont("04b.gdf");
imagestring($im, $font, 0, 0, "Hello", $black);
imagepng($im);
?>
```

Voir aussi `imagefontwidth` et `imagefontheight`.

8.53.76 `imagepalettecopy()` : Copie la palette d'une image à l'autre

void `imagepalettecopy` (resource destination , resource source)

`imagepalettecopy` copie la palette de l'image source dans l'image destination .

8.53.77 `imagepng()` : Envoie une image PNG vers un navigateur ou un fichier

bool `imagepng` (resource image , string filename)

`imagepng` envoie l'image GD (image) au format PNG sur la sortie standard (typiquement, le navigateur web), ou si filename est fourni, l'envoi dans un fichier.

Exemple avec `imagepng`

```
<?php
$im = imagecreatefrompng("test.png");
imagepng($im);
?>
```

Le nom du fichier est optionnel et, dans ce cas, l'image sera transmise directement à la sortie standard. En envoyant une image de type image/png content-type (grâce à la fonction `header`), vous pouvez créer des images PNG avec des scripts PHP.

Voir aussi [imagegif](#) , [imagewbmp](#) , [imagejpeg](#) et [imagetypes](#) .

8.53.78 imagepolygon() : Dessine un polygone

bool **imagepolygon** (resource image , array points , int num_points , int color)

[imagepolygon](#) dessine un polygone dans l'image image . points est un tableau PHP qui contient les sommets du polygone sous la forme : points[0] = x0, points[1] = y0, points[2] = x1, points[3] = y1, etc. num_points est le nombre de sommets. num_points est le nombre de points (sommets).

Exemple avec [imagepolygon](#)

```
<?php
// nouvelle image
$image = imagecreatetruecolor(400,300);

// couleur de fond
$bg = imagecolorallocate($image,0,0,0);

// couleur pour les bords du polygone
$col_poly = imagecolorallocate($image,255,255,255);

// on dessine le polygone
imagepolygon($image,
    array (
        0, 0,
        100, 200,
        300, 200
    ),
    3,
    $col_poly);

// on affiche l'image
header("Content-type: image/png");
imagepng($image);

?>
```

Voir aussi [imagecreate](#) et [imagecreatetruecolor](#) .

8.53.79 imagepsbbox() : Retourne le rectangle entourant un texte et dessiné avec une police PostScript Type1

array **imagepsbbox** (string text , int font , int size , int space , int tightness , float angle)

size est exprimé en pixels.

space permet de changer la valeur par défaut du caractère espace. Cette valeur est ajoutée lors des dessins et, donc, peut être négative.

tightness permet de contrôler la quantité d'espace entre les caractères. Cette quantité est ajoutée lors des dessins, et peut donc être négative.

angle est en degrés.

Les paramètres space et tightness sont exprimés en unité d'espacement de caractères, avec 1 unité vaut 1/1000 d'un em carré (un em : unité de mesure représentée par un carré dont la

dimension horizontale est la même que le corps du caractère).

Les paramètres `space` , `tightness` et `angle` sont optionnels.

Le rectangle entourant est calculé en utilisant les informations disponibles sur les tailles de caractères, et, malheureusement, il a tendance à être légèrement différent du résultat réel final. Si l'angle est de 0 degré, vous pouvez-vous attendre à avoir besoin d'un rectangle d'au moins un pixel plus grand dans toutes les directions.

Note
Cette fonction n'est disponible que si PHP est compilé en utilisant <code>--enable-t1lib[=DIR]</code> .

`imagepsbbox` retourne un tableau contenant les éléments suivants :

0	Abscisse inférieure gauche
1	Ordonnée inférieure gauche
2	Abscisse supérieure droite
3	Ordonnée supérieure droite

Voir aussi [imagepstext](#) .

8.53.80 `imagepscopyfont()` : Fait une copie d'une police chargée pour modification ultérieure

int `imagepscopyfont` (resource fontindex)

Utilisez cette fonction si vous devez faire des modifications dans une police, comme l'extension ou la condensation ou encore, modifier son vecteur d'encodage, tout en conservant l'original. Notez que la police que vous copiez doit être obtenue par [imagepsloadfont](#) , et non pas une police qui a elle-même été copiée. Vous pouvez aussi faire des modifications avant de la copier.

Si vous utilisez cette fonction, vous **DEVEZ** libérer les polices obtenues vous-même, et dans l'ordre inverse. Sinon, votre script sera **suspendu** .

Si tout a bien été fait, un index valide de police sera retourné et pourra être utilisé ultérieurement. Sinon, la fonction retournera FALSE et affichera un message décrivant l'erreur.

Note
Cette fonction n'est disponible que si PHP est compilé en utilisant <code>--enable-t1lib[=DIR]</code> .

Voir aussi [imagepsloadfont](#) .

8.53.81 `imagepsencodefont()` : Change le codage vectoriel d'un caractère dans une police

bool `imagepsencodefont` (resource font_index , string encodingfile)

`imagepsencodefont` charge le codage vectoriel d'un caractère depuis un fichier et change le codage vectoriel de la police correspondante. Etant donné que les polices PostScript ne disposent pas des caractères au-delà de 127, vous aurez sûrement besoin de les changer si vous utilisez une autre langue que l'anglais. Le format exact est décrit dans la documentation T1libs. T1lib est disponible en deux formes : `IsoLatin1.enc` et `IsoLatin2.enc`.

Si vous commencez à utiliser cette fonction régulièrement, une meilleure solution est de définir un encodage, et de l'utiliser avec `ps.default_encoding` dans le fichier de configuration pour utiliser par défaut l'encodage correct.

Note

Cette fonction n'est disponible que si PHP est compilé en utilisant <code>--enable-t1lib[=DIR]</code> .

8.53.82 `imagepsextextfont()` : Etend ou condense une police de caractères

bool **imagepsextextfont** (int font_index , float extend)

`imagepsextextfont` étend ou condense la police de caractères font . Si la valeur de extend est inférieure à 1, ce sera une condensation.

Note

Cette fonction n'est disponible que si PHP est compilé en utilisant <code>--enable-t1lib[=DIR]</code> .

8.53.83 `imagepsfreefont()` : Libère la mémoire occupée par une police PostScript Type 1

bool **imagepsfreefont** (resource fontindex)

`imagepsfreefont` libère la mémoire occupée par une police PostScript Type 1.

Note

Cette fonction n'est disponible que si PHP est compilé en utilisant <code>--enable-t1lib[=DIR]</code> .

Voir aussi `imagepsloadfont` .

8.53.84 `imagepsloadfont()` : Charge une police PostScript Type 1 depuis un fichier

resource **imagepsloadfont** (string filename)

En cas de succès, `imagepsloadfont` retourne un index de police, qui pourra être utilisé pour des opérations ultérieures. Sinon, `imagepsloadfont` retourne FALSE et affiche un message décrivant ce qui est arrivé.

Exemple avec <code>imagepsloadfont</code>

<pre><?php header ("Content-type: image/jpeg"); \$im = imagecreatetruecolor (350, 45); \$black = imagecolorallocate (\$im, 0, 0, 0); \$white = imagecolorallocate (\$im, 255, 255, 255); \$font = imagepsloadfont ("bchbi.pfb"); // ou bien situez un de vos fichiers .pfb imagepstext (\$im, "Test ... Ca marche!", \$font, 32, \$white, \$black, 32, 32); imagepsfreefont (\$font); imagejpeg (\$im, "", 100); //Meilleure qualité... imagedestroy (\$im); ?></pre>

Note

Cette fonction n'est disponible que si PHP est compilé en utilisant `--enable-t1lib[=DIR]` .

Voir aussi [imagepsfreefont](#) .

8.53.85 `imagepslantfont()` : Incline une police de caractères PostScript

bool **imagepslantfont** (resource font_index , float slant)

`imagepslantfont` met en italique la police de caractères font avec le coefficient slant .

Note

Cette fonction n'est disponible que si PHP est compilé en utilisant `--enable-t1lib[=DIR]` .

8.53.86 `imagepstext()` : Dessine un texte sur une image avec une police PostScript Type1

array **imagepstext** (resource image , string text , resource font , int size , int foreground , int background , int x , int y , *int space* , int tightness , float angle , int antialias_steps)

foreground est la couleur avec laquelle le texte va être dessiné. background est la couleur d'anti aliasing. Aucun pixel avec la couleur background n'est dessiné, ce qui fait que l'arrière-plan n'a pas besoin d'être dans une couleur fixe.

Les coordonnées données (x , y) définissent l'origine du premier caractère (grossièrement, le coin inférieur gauche du caractère). Ceci est différent de la fonction `imagestring` , où (x, y) définissait le coin supérieur gauche du premier caractère. Reportez-vous à la documentation PostScript pour avoir des détails à propos des polices et de leurs tailles.

space permet de changer la taille par défaut du caractère d'espacement. Cette valeur peut être négative.

tightness permet de contrôler la quantité d'espace entre deux caractères. Cette valeur peut être négative.

angle est en degrés.

size est exprimé en pixels.

antialias_steps permet de contrôler le nombre de couleurs du texte anti-aliasé. Les valeurs autorisées sont 4 et 16. 16 est recommandée pour les polices de moins de 20 pixels, car l'effet est alors visible. Avec les tailles plus grandes, utilisez de préférence 4, qui est moins gourmande en ressources.

Les paramètres space et tightness sont exprimés en unité d'espaces caractère, ce qui vaut 1/1000ème d'un em-carré (un em : unité de mesure représentée par un carré dont la dimension horizontale est la même que le corps du caractère).

Les paramètres space , tightness , angle et antialias sont optionnels.

`imagepstext` retourne un tableau contenant les éléments suivants :

0	Abcisse inférieure gauche
1	Ordonnée inférieure gauche

2	Abcisse supérieure droite
3	Ordonnée supérieure droite
Note	
Cette fonction n'est disponible que si PHP est compilé en utilisant --enable-t1lib[=DIR] .	

Voir aussi [imagepsbbox](#) .

8.53.87 imagerectangle() : Dessine un rectangle

bool **imagerectangle** (resource image , int x1 , int y1 , int x2 , int y2 , int col)

imagerectangle dessine un rectangle dans la couleur color , dans l'image image , en commençant au point supérieur gauche (x1 , y1), et en finissant au point inférieur droit (x2 , y2). Le coin supérieur gauche est l'origine (0,0).

8.53.88 imagerotate() : Fait tourner une image d'un angle

resource **imagerotate** (resource src_im , float angle , int bgd_color , int ignore_transparent)

imagerotate fait tourner l'image src_im d'un angle de angle degrés. bgd_color spécifie la couleur des zones qui seront découvertes après la rotation.

Le centre de rotation est le centre de l'image, et l'image tournée est réduite de telle sorte que les ajustements apportés à l'image tournée - les bords ne soient pas coupés.

Si ignore_transparent est défini et ne vaut pas zéro, les couleurs transparentes seront ignorées. Ce paramètre a été ajouté en PHP 5.1.

Rotation d'une image de 180 degrés
Cet exemple tourne une image de 180 degrés - à l'envers.
<pre>// Fichier et degrés de rotation \$filename = 'test.jpg'; \$degrees = 180; // Content type header('Content-type: image/jpeg'); // Chargement \$source = imagecreatefromjpeg(\$filename); // Rotation \$rotate = imagerotate(\$source, \$degrees, 0); // Affichage imagejpeg(\$rotate);</pre>
Note
Cette fonction n'est disponible que si PHP est compilé avec la version embarquée de la bibliothèque GD.

8.53.89 imagesavealpha() : Configure l'enregistrement des informations complètes du canal alpha lors de sauvegardes d'images PNG

bool **imagesavealpha** (resource image , bool saveflag)

imagesavealpha définit l'option pour essayer de sauvegarder toutes les informations du canal alpha (en opposition à la transparence à couleur unique) lors de la sauvegarde d'images PNG.

Vous devez désactiver le alphablending (imagealphablending (\$im, FALSE)) pour l'utiliser.

Le canal alpha n'est pas supporté par tous les navigateurs ; si vous avez des problèmes avec le vôtre, essayez de charger le script avec un navigateur compatible avec les canaux alpha, c'est à dire le dernier Mozilla.

Note

Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.

Voir aussi <u>imagealphablending</u> .
--

8.53.90 imagesetbrush() : Modifie la brosse pour le dessin des lignes

bool **imagesetbrush** (resource image , resource brush)

imagesetbrush remplace la brosse courante pour le dessin des lignes par brush . Cette brosse sera alors utilisée avec des fonctions comme imageline ou imagepolygon et avec les couleurs spéciales IMG_COLOR_BRUSHED ou IMG_COLOR_STYLED BRUSHED .

Note

Vous n'avez rien à faire lorsque vous en avez terminé avec une brosse, mais si vous détruisez l'image de brosse, vous ne DEVEZ plus utiliser les options IMG_COLOR_BRUSHED et IMG_COLOR_STYLED BRUSHED des fonctions <u>imageline</u> et <u>imagepolygon</u> , avant d'avoir créé une nouvelle brosse.
--

Note

<u>imagesetbrush</u> a été ajoutée en PHP 4.0.6.
--

8.53.91 imagesetpixel() : Dessine un pixel

bool **imagesetpixel** (resource image , int x , int y , int color)

imagesetpixel dessine un pixel au point (x , y) (le coin supérieur gauche est l'origine (0,0)) dans l'image image , et avec la couleur color .

Voir aussi imagecreatetruecolor et imagecolorallocate .

8.53.92 imagesetstyle() : Configure le style pour le dessin des lignes

bool **imagesetstyle** (resource image , array style)

imagesetstyle permet de choisir le style à utiliser lors du dessin des lignes (comme avec les fonctions imageline et imagepolygon) lors de l'utilisation de la couleur spéciale IMG_COLOR_STYLED ou bien lors du dessin de lignes avec la couleur IMG_COLOR_STYLED BRUSHED .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre style est un tableau de pixels. L'exemple suivant dessine une ligne pointillée depuis le coin supérieur gauche vers le coin inférieur droit de l'image :

Exemple avec `imagesetstyle`

```
<?php
header ("Content-type: image/jpeg");
$im = imagecreatetruecolor (100, 100);
$w  = imagecolorallocate ($im, 255, 255, 255);
$red = imagecolorallocate ($im, 255, 0, 0);

/* Dessine une ligne pointillée de 5 pixels rouges, 5 pixels blancs */
$style = array ($red,$red,$red,$red,$red,$w,$w,$w,$w,$w);
imagesetstyle ($im, $style);
imageline ($im, 0, 0, 100, 100, IMG_COLOR_STYLED);

/* Dessine une ligne avec des smileys, en utilisant imagesetbrush() et imagesetstyle */
$style = array ($w,$w,$w,$w,$w,$w,$w,$w,$w,$w,$w,$red);
imagesetstyle ($im, $style);

$brush = imagecreatefrompng ("http://www.libpng.org/pub/png/images/smile.happy.png");
$w2 = imagecolorallocate($brush,255,255,255);
imagecolortransparent ($brush, $w2);
imagesetbrush ($im, $brush);
imageline ($im, 100, 0, 0, 100, IMG_COLOR_STYLED BRUSHED);

imagejpeg ($im);
imagedestroy ($im);
?>
```

Note

Cette fonction a été ajoutée en PHP 4.0.6.

Voir aussi [imagesetbrush](#) et [imageline](#) .

8.53.93 `imagesetthickness()` : Modifie l'épaisseur d'un trait

bool `imagesetthickness` (resource image , int thickness)

`imagesetthickness` modifie l'épaisseur du trait des lignes de l'image image . Cette épaisseur intervient dans les dessins de polygones, ellipses, cercles, rectangles, etc. thickness est en pixels.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.

8.53.94 `imagesettile()` : Modifie l'image utilisée pour le carrelage

bool `imagesettile` (resource image , resource tile)

`imagesettile` remplace l'image de pavage courante par l'image tile , à utiliser dans tous les remplissages (comme avec les fonctions [imagefill](#) et [imagefilledpolygon](#)) lors des remplissages avec l'option IMG_COLOR_TILED .

Une image de carrelage est une image utilisée pour remplir une zone, de manière répétitive. N'importe quelle image GD peut servir d'image de remplissage. L'utilisation de la couleur transparente (gérée avec la fonction [imagecolortransparent](#)) permet à certaines zones d'apparaître

à travers le carrelage.

Note

Vous n'avez rien à faire lorsque vous en avez terminé avec une brosse, mais si vous détruisez l'image de brosse, vous ne DEVEZ plus utiliser l'option `IMG_COLOR_TILED` des fonctions `imagefill` et `imagefilledpolygon`, avant d'avoir créé une nouvelle brosse.

8.53.95 imagestring() : Dessine une chaîne horizontale

bool **imagestring** (resource image , int font , int x , int y , string s , int col)

`imagestring` dessine une chaîne sur une ligne horizontale, dans l'image `image`, avec le coin en haut à gauche aux coordonnées (`x` , `y`) (le coin supérieur gauche est l'origine (0,0)) avec la couleur `color`. Si l'argument de police vaut 1, 2, 3, 4 ou 5, une des polices par défaut sera utilisée).

Exemple avec `imagestring`

```
<?php
// nouvelle image 100*30
$im = imagecreatetruecolor(100, 30);

// fond blanc et texte bleu
$bg = imagecolorallocate($im, 255, 255, 255);
$textcolor = imagecolorallocate($im, 0, 0, 255);

// ajout de la phrase en haut à gauche
imagestring($im, 5, 0, 0, "Bonjour le monde !", $textcolor);

// affichage de l'image
header("Content-type: image/jpeg");
imagejpeg($im);
?>
```

Voir aussi `imageloadfont` et `imagefttext`.

8.53.96 imagestringup() : Dessine une chaîne verticale

bool **imagestringup** (resource image , int font , int x , int y , string s , int col)

`imagestringup` dessine une chaîne sur une ligne verticale dans l'image `image` aux coordonnées (`x` , `y`) (l'origine est le coin supérieur gauche (0,0)) dans la couleur `color`. Si la police utilisée est 1, 2, 3, 4 ou 5, une police par défaut sera utilisée.

Voir aussi `imageloadfont`.

8.53.97 imagesx() : Retourne la largeur d'une image

int **imagesx** (resource image)

`imagesx` retourne la largeur de l'image référencée par `image`.

Exemple avec `imagesx`

```
<?php
// Création d'une image 300*200 image
```

```

$img = imagecreatetruecolor(300, 200);

echo imagesx($img); // 300

?>

```

Voir aussi [getimagesize](#) , [imagecreatetruecolor](#) et [imagesy](#) .

8.53.98 imagesy() : Retourne la hauteur de l'image

int **imagesy** (resource image)

[imagesy](#) retourne la hauteur de l'image référencée par image .

Exemple avec [imagesy](#)

```

<?php

// Création d'une image 300*200
$img = imagecreatetruecolor(300, 200);

echo imagesy($img); // 200

?>

```

Voir aussi [getimagesize](#) , [imagecreatetruecolor](#) et [imagesx](#) .

8.53.99 imagetruecolortopalette() : Convertit une image en vraies couleurs en image à palette

bool **imagetruecolortopalette** (resource image , bool dither , int ncolors)

[imagetruecolortopalette](#) convertit l'image en vraies couleurs image en image à palette. Le code de cette fonction est directement tiré de la bibliothèque du Independent JPEG Group , qui est tout simplement génial. Le code a été modifié pour préserver l'essentiel du canal alpha dans la nouvelle palette, en plus de conserver les couleurs du mieux possible. Mais cela ne fonctionne pas toujours comme voulu. Il est alors préférable de générer un résultat en vraies couleurs, ce qui donne toujours le meilleur rendu.

Si dither vaut TRUE , cela active le dither : l'image sera un peu plus granuleuse, mais l'approximation des couleurs sera meilleure.

ncolors est le nombre maximal de couleurs dans la palette finale.

Note

Cette fonction requiert la bibliothèque GD 2.0.1 ou supérieure.

8.53.100 imagettfbbox() : Retourne le rectangle entourant un texte et dessiné avec une police TrueType

array **imagettfbbox** (float size , float angle , string fontfile , string text)

[imagettfbbox](#) calcule et retourne le rectangle entourant le texte text , écrit avec une police truetype.

size	La taille de la police en pixel.
angle	L'angle en degrés dans lequel le paramètre text sera mesuré.
fontfile	Le nom de la police TrueType (peut être une URL). Suivant la version de la bibliothèque GD utilisée par PHP, ce paramètre peut chercher des fichiers qui ne commence pas par un slash (" / ") de fin mais plutôt .ttf et cherchera tout le long des chemins de fonts définis.
text	La chaîne à mesurer.

[imaggotbbox](#) retourne un tableau avec 8 éléments représentant les 4 sommets du rectangle ainsi défini.

0	Coin inférieur gauche, abscisse
1	Coin inférieur gauche, ordonnée
2	Coin inférieur droit, abscisse
3	Coin inférieur droit, ordonnée
4	Coin supérieur droit, abscisse
5	Coin supérieur droit, ordonnée
6	Coin supérieur gauche, abscisse
7	Coin supérieur gauche, ordonnée

Les positions des points sont relatives au texte **text**, indépendamment de l'angle : coin supérieur gauche faire référence au coin supérieur gauche du texte écrit horizontalement.

[imaggotbbox](#) requiert les bibliothèques GD et Freetype.

Voir aussi [imaggottext](#).

8.53.101 [imaggottext\(\)](#) : Dessine un texte avec une police TrueType

array **[imaggottext](#)** (resource image , float size , float angle , int x , int y , int color , string fontfile , string text)

image	La ressource de l'image. Voir la fonction imagecreatetruecolor .
size	La taille de la police. En fonction de la version de votre bibliothèque GD, elle doit être spécifiée en pixels (GD1) ou en points (GD2).
angle	L'angle, en degrés ; 0 degré correspond à la lecture du texte de gauche à droite. Les valeurs positives représentent une rotation dans le sens contraire des aiguilles d'une montre. Par exemple, une valeur de 90 correspondra à une lecture du texte de bas en haut.
x	Les coordonnées données par x et y définiront la position du premier caractère (le coin bas-gauche du caractère). Cela est différent de la fonction imagestring , où x et y définissent le coin haut-gauche du premier caractère. Par exemple, "haut gauche" correspond à 0, 0.
y	

L'ordonnée Y. Cela définit la position de la ligne de base de la police, et non pas le bas des caractères.

color

L'index de la couleur. Utiliser un index de couleur négatif désactivera l'antialiasing. Voir la fonction [imagecolorallocate](#) .

fontfile

Le chemin vers la police TrueType que vous souhaitez utiliser.

Dépendant de la version de la bibliothèque GD utilisée par PHP, **lorsque fontfile ne commence pas par un caractère / alors .ttf sera ajouté** au nom du fichier et la bibliothèque recherchera ce nom de fichier dans les dossiers de police, définis par la bibliothèque.

Lorsque vous utilisez une version de la bibliothèque GD inférieure à 2.0.18, un caractère espace , plutôt qu'un point-virgule, est utilisé pour définir des chemins alternatifs pour les fichiers de police. Si vous utilisez involontairement cette fonctionnalité, vous aurez ce message d'erreur : Warning: Could not find/open font . Pour ces versions affectées, la seule solution est de déplacer la police à un endroit qui ne contient pas d'espaces dans les noms de dossier.

Dans le cas fréquent où une police réside dans le même dossier que le script l'utilisant, l'astuce suivante vous épargnera tous les problèmes.

```
<?php
// Définition de la variable d'environnement pour GD
putenv('GDFONTPATH=' . realpath('.'));

// Nom de la police à utiliser (notez l'absence de l'extension .ttf)
$font = 'SomeFont';
?>
```

text

La chaîne de texte.

Peut inclure des références à des caractères numériques, décimales (sous la forme : €) pour accéder aux caractères d'une police au delà du premier 127. Les chaînes de caractères encodées en UTF-8 peuvent être passées directement.

Si un caractère est utilisé dans une chaîne qui n'est pas supportée par la police, un rectangle creux remplacera le caractère.

[imagettftext](#) retourne un tableau de 8 éléments représentant quatre points marquant les limites du texte. L'ordre des points est : inférieur gauche, inférieur droit, supérieur droit, supérieur gauche. Les points sont relatifs au texte par rapport à l'angle, donc, "supérieur gauche" signifie dans le coin en haut à gauche lorsque vous regardez le texte horizontalement.

Exemple avec [imagettftext](#)

Cet exemple produira une image PNG blanche de 400x30 pixels, avec le texte "Test..." en noir, avec une ombre grise, utilisant la police Arial.

```
<?php
// Définition du content-type
header("Content-type: image/png");

// Création de l'image
$im = imagecreatetruecolor(400, 30);
```

```
// Création de quelques couleurs
$white = imagecolorallocate($im, 255, 255, 255);
$grey = imagecolorallocate($im, 128, 128, 128);
$black = imagecolorallocate($im, 0, 0, 0);

// Le texte à dessiner
$text = 'Test...';
// Remplacez le chemin par votre propre chemin de police
$font = 'arial.ttf';

// Ajout d'ombres au texte
imagefttext($im, 20, 0, 11, 21, $grey, $font, $text);

// Ajout du texte
imagefttext($im, 20, 0, 10, 20, $black, $font, $text);

// Utiliser imagepng() donnera un texte plus claire,
// comparé à l'utilisation de la fonction imagejpeg()
imagepng($im);
imagedestroy($im);
?>
```

Cette fonction requiert la bibliothèque GD ainsi que la bibliothèque [FreeType](#) .

Voir aussi [imageftbbox](#) .

8.53.102 imagetypes() : Retourne les types d'images supportés par la version courante de PHP

int **imagetypes** (void)

[imagetypes](#) retourne un champ de bits correspondant aux formats d'images supportés par la version de GD utilisée. Les valeurs suivantes sont possibles : IMG_GIF | IMG_JPG | IMG_PNG | IMG_WBMP | IMG_XPM . Pour vous assurer du support PNG , faites ceci :

Exemple avec [imagetypes](#)

```
<?php
if (imagetypes() & IMG_PNG) {
    echo "Le type PNG est supporté";
}
?>
```

8.53.103 imagewbmp() : Affiche une image WBMP

bool **imagewbmp** (resource image , string filename , int foreground)

[imagewbmp](#) crée l'image WBMP dans le fichier filename , à partir de l'image image . Le paramètre image a été créé avec la fonction [imagecreatetruecolor](#) .

filename est optionnel et, s'il est omis, l'image sera envoyée directement au client. En plaçant l'en-tête IMAGE/VND.WAP.WBMP , dans le champ "content-type", vous pourrez afficher une image WBMP .

Note

Le support WBMP n'est disponible que si PHP a été compilé avec GD-1.8 ou plus récent.

En passant le paramètre optionnel `foreground` , vous pouvez choisir la couleur de fond. Utilisez l'identifiant retourné par [imagecolorallocate](#) comme valeur de ce paramètre. La couleur de fond par défaut est noire.

Voir aussi [image2wbmp](#) , [imagepng](#) , [imagegif](#) , [imagejpeg](#) et [imagetypes](#) .

8.53.104 `imagexbm()` : Génère une image au format XBM

`bool imagexbm (resource image , string filename , int foreground)`

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.
Note
Cette fonction n'est disponible que si PHP est compilé avec la version embarquée de la bibliothèque GD.

8.53.105 `iptcembed()` : Intègre des données binaires IPTC dans une image JPEG

`mixed iptcembed (string iptcdata , string jpeg_file_name , int spool)`

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.53.106 `iptcparse()` : Analyse un bloc binaire IPTC <http://www.iptc.org/> et recherche les balises simples

`array iptcparse (string iptcblock)`

[iptcparse](#) analyse un bloc binaire IPTC et recherche les balises simples. [iptcparse](#) retourne un tableau avec les balises comme index et les valeurs de ces balises IPTC dans les valeurs de tableau correspondantes. En cas d'erreur, ou si aucune balise IPTC n'a été trouvée, cette fonction retourne FALSE . Voir [getimagesize](#) pour un exemple.

8.53.107 `jpeg2wbmp()` : Convertit une image JPEG en image WBMP

`int jpeg2wbmp (string jpegname , string wbmname , int d_height , int d_width , int threshold)`

[jpeg2wbmp](#) convertit l'image JPEG du fichier `jpegname` au format WBMP, et la sauve dans le fichier `wbmname` . Les paramètres `d_height` et `d_width` vous permettent de spécifier la hauteur et la largeur (respectivement) de l'image de destination.

Note
Le support JPEG n'est disponible que si PHP a été compilé avec GD-1.8 ou supérieur.
Note
Le support WBMP n'est disponible que si PHP a été compilé avec GD-1.8 ou supérieur.

Voir aussi [png2wbmp](#) .

8.53.108 **png2wbmp()** : Convertit une image PNG en image WBMP

int **png2wbmp** (string pngname , string wbmpname , int d_height , int d_width , int threshold)

png2wbmp convertit l'image PNG du fichier pngname au format WBMP, et la sauve dans le fichier wbmpname . Les paramètres d_height et d_width vous permettent de spécifier la hauteur et la largeur (respectivement) de l'image de destination.

Note

Le support WBMP n'est disponible que si PHP a été compilé avec GD-1.8 ou supérieur.

Voir aussi jpeg2wbmp .

8.54 IMAP

8.54.1 Introduction

Ces fonctions ne sont pas limitées au protocole IMAP , malgré leur nom. La bibliothèque cliente C supporte aussi les méthodes d'accès à NNTP , POP3 et aux boîtes aux lettres locales.

8.54.2 Pré-requis

Cette extension requiert la bibliothèque C cliente. Téléchargez cette dernière version à <ftp://ftp.cac.washington.edu/imap/> et compilez-la.

Il est important de ne pas copier les fichiers sources IMAP directement dans le dossier d'inclusion système sous peine de conflits. A la place de cela, créez un nouveau dossier dans le dossier d'inclusion système, comme /usr/local/imap-2000b/ (le chemin et le nom dépendent de votre configuration et de votre version d'IMAP) et dans ce nouveau dossier, créez les dossiers nommés lib/ et include/ . Depuis le dossier c-client des sources IMAP, copiez tous les fichiers *.h dans le dossier include/ et tous les fichiers *.c dans le dossier lib/ . Additionnellement, lorsque vous compilez IMAP, un fichier nommé c-client.a est crée. Mettez le également dans le dossier lib/ mais renommez le en libc-client.a .

Note
Pour compiler la bibliothèque C cliente avec SSL et/ou avec le support Kerberos, lisez la doc fournie dans la distribution.
Note
Sur Mandrake Linux, la bibliothèque IMAP (libc-client.a) est compilée sans le support Kerberos. Une version séparée avec SSL (client-PHP4.a) est installée. La bibliothèque doit être recompilée pour ajouter le support Kerberos.

8.54.3 Installation

Pour avoir accès à ces fonctions, vous devez compiler PHP avec l'option --with-imap[=DIR] , où DIR est le préfixe de l'installation du client C. Dans nos exemples, vous pourriez utiliser --with-imap=/usr/local/imap-2000b . Le nom du dossier d'installation dépend de l'emplacement où vous avez créé le dossier, en fonction de la description ci-dessus. Les utilisateurs Windows peuvent inclure la bibliothèque DLL php_imap.dll dans le php.ini . IMAP n'est pas supporté sur les systèmes plus anciens que Windows 2000. Cela est dû au fait que IMAP utilise les fonctions de cryptage pour activer les connexions SSL aux serveurs de mails.

Note
Suivant la configuration de la bibliothèque C cliente, vous pouvez aussi avoir besoin d'ajouter --with-imap-ssl=/path/to/openssl/ et/ou --with-kerberos=/path/to/kerberos dans la ligne de configuration PHP.
Attention
L'extension <u>IMAP</u> ne peut être utilisée simultanément avec l'extension <u>recode</u> , l'extension <u>YAZ</u> ou l'extension <u>Cyrus</u> . Ceci est dû au fait que ces bibliothèques utilisent un symbole interne commun.

8.54.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.54.5 Types de ressources

8.54.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

NIL (entier)
OP_DEBUG (entier)
OP_READONLY (entier)
Ouvre une boîte aux lettres en lecture seule
OP_ANONYMOUS (entier)
Ne pas utiliser, ou modifier le fichier .newsrsrc pour les news, (NNTP uniquement).
OP_SHORTCACHE (entier)
OP_SILENT (entier)
OP_PROTOTYPE (entier)
OP_HALFOPEN (entier)
Pour les noms IMAP et NNTP, ouvre une connexion mais n'ouvre pas une boîte aux lettres.
OP_EXPUNGE (entier)
OP_SECURE (entier)
CL_EXPUNGE (entier)
purger automatiquement la boîte aux lettres lors de l'appel de imap_close
FT_UID (entier)
Le paramètre est un UID.
FT_PEEK (entier)
Ne pas lever le drapeau \Seen (Message lu) s'il n'est pas déjà levé.
FT_NOT (entier)
FT_INTERNAL (entier)
La chaîne renvoyée est au format interne, et ne va pas canoniser les CRLF.
FT_PREFETCHTEXT (entier)
ST_UID (entier)
la séquence contient des UID au lieu de numéros de séquence
ST_SILENT (entier)
ST_SET (entier)
CP_UID (entier)
La séquence de nombres contient des UID
CP_MOVE (entier)
Efface les messages après copie avec imap_mail_copy
SE_UID (entier)
Retourne des UID à la place de numéros
SE_FREE (entier)
SE_NOPREFETCH (entier)
Ne pas pré-télécharger les messages trouvés
SO_FREE (entier)
SO_NOSERVER (entier)
SA_MESSAGES (entier)
SA_RECENT (entier)

SA_UNSEEN (entier)
SA_UIDNEXT (entier)
SA_UIDVALIDITY (entier)
SA_ALL (entier)
LATT_NOINFERIORS (entier)
 Cette boîte aux lettres n'a pas d'"enfants" (il n'y a plus de boîtes aux lettres en dessous de celle-ci).
LATT_NOSELECT (entier)
 Ceci est juste un container, pas une boîte aux lettres (vous ne pouvez pas l'ouvrir).
LATT_MARKED (entier)
 Cette boîte aux lettres est marquée. Utilisé uniquement avec UW-IMAPD.
LATT_UNMARKED (entier)
 Cette boîte aux lettres n'est pas marquée. Utilisé uniquement avec UW-IMAPD.
SORTDATE (entier)
 Critère de tri pour imap_sort : Date du message
SORTARRIVAL (entier)
 Critère de tri pour imap_sort : Date d'arrivée
SORTFROM (entier)
 Critère de tri pour imap_sort : Nom de la première boîte aux lettres de l'adresse d'origine (From address)
SORTSUBJECT (entier)
 Critère de tri pour imap_sort : Sujet du message
SORTTO (entier)
 Critère de tri pour imap_sort : Nom de la première boîte aux lettres de destination (To address)
SORTCC (entier)
 Critère de tri pour imap_sort : Nom de la boîte aux lettres de copie cachée (cc address)
SORTSIZE (entier)
 Critère de tri pour imap_sort : Taille du message en octets
TYPETEXT (entier)
TYPEMULTIPART (entier)
TYPEMESSAGE (entier)
TYPEAPPLICATION (entier)
TYPEAUDIO (entier)
TYPEIMAGE (entier)
TYPEVIDEO (entier)
TYPEOTHER (entier)
ENC7BIT (entier)
ENC8BIT (entier)
ENCBINARY (entier)
ENCBASE64 (entier)
ENCQUOTEDPRINTABLE (entier)
ENCOTHER (entier)

8.54.7 Voir aussi

Ce document ne peut entrer dans les détails de tous les sujets abordés. Plus d'informations sont disponibles avec la documentation de la bibliothèque C (docs/internal.txt) ainsi que les RFC suivantes :

- [RFC2821](#) : Simple Mail Transfer Protocol (SMTP).
- [RFC2822](#) : Standard for ARPA internet text messages.

- [RFC2060](#) : Internet Message Access Protocol (IMAP) Version 4rev1.
- [RFC1939](#) : Post Office Protocol Version 3 (POP3).
- [RFC977](#) : Network News Transfer Protocol (NNTP).
- [RFC2076](#) : Common Internet Message Headers.
- [RFC2045](#) , [RFC2046](#) , [RFC2047](#) , [RFC2048](#) et [RFC2049](#) : Multipurpose Internet Mail Extensions (MIME).

Une étude approfondie est aussi disponible dans les livres suivants (en anglais): [Programming Internet Email](#) par David Wood et [Managing IMAP](#) par Dianna Mullet & Kevin Mullet.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Voir aussi](#)
- [imap_8bit](#)
- [imap_alerts](#)
- [imap_append](#)
- [imap_base64](#)
- [imap_binary](#)
- [imap_body](#)
- [imap_bodystruct](#)
- [imap_check](#)
- [imap_clearflag_full](#)
- [imap_close](#)
- [imap_createmailbox](#)
- [imap_delete](#)
- [imap_deletemailbox](#)
- [imap_errors](#)
- [imap_expunge](#)
- [imap_fetch_overview](#)
- [imap_fetchbody](#)
- [imap_fetchheader](#)
- [imap_fetchstructure](#)
- [imap_get_quota](#)
- [imap_get_quotaroot](#)
- [imap_getacl](#)
- [imap_getmailboxes](#)
- [imap_getsubscribed](#)
- [imap_header](#)
- [imap_headerinfo](#)
- [imap_headers](#)
- [imap_last_error](#)
- [imap_list](#)
- [imap_listmailbox](#)
- [imap_listscan](#)
- [imap_listsubscribed](#)
- [imap_lsub](#)
- [imap_mail_compose](#)
- [imap_mail_copy](#)
- [imap_mail_move](#)
- [imap_mail](#)

- [imap_mailboxmsginfo](#)
- [imap_mime_header_decode](#)
- [imap_msgno](#)
- [imap_num_msg](#)
- [imap_num_recent](#)
- [imap_open](#)
- [imap_ping](#)
- [imap_qprint](#)
- [imap_renamemailbox](#)
- [imap_reopen](#)
- [imap_rfc822_parse_adrlist](#)
- [imap_rfc822_parse_headers](#)
- [imap_rfc822_write_address](#)
- [imap_scanmailbox](#)
- [imap_search](#)
- [imap_set_quota](#)
- [imap_setacl](#)
- [imap_setflag_full](#)
- [imap_sort](#)
- [imap_status](#)
- [imap_subscribe](#)
- [imap_thread](#)
- [imap_timeout](#)
- [imap_uid](#)
- [imap_undelete](#)
- [imap_unsubscribe](#)
- [imap_utf7_decode](#)
- [imap_utf7_encode](#)
- [imap_utf8](#)

8.54.9 [imap_alerts\(\)](#) : Retourne toutes les alertes

array [imap_alerts](#) (void)

[imap_alerts](#) retourne tous les messages d'alerte IMAP générés depuis le dernier appel à [imap_alerts](#) ou depuis le début de la page. Lorsque [imap_alerts](#) est appelée, la pile d'alertes est vidée.

8.54.10 [imap_append\(\)](#) : Ajoute une chaîne dans une boîte aux lettres

bool [imap_append](#) (resource [imap_stream](#) , string [mbox](#) , string [message](#) , *string options*)

[imap_append](#) ajoute un message dans la boîte aux lettres [mbox](#) . Si l'argument [options](#) est utilisé, [options](#) sera aussi écrit dans la boîte aux lettres.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Lors des échanges avec le serveur Cyrus IMAP, vous devrez utiliser "\r\n" comme terminaison de ligne, à la place de "\n" ou l'opération échouera.

Exemple avec [imap_append](#)

```
<?php
```

```

$stream = imap_open("{votre.hote.imap}INBOX.Drafts", "username", "password");

$check = imap_check($stream);
echo "Nombre de message avant ajout : ". $check->Nmsgs . "\n";

imap_append($stream, "{votre.hote.imap}INBOX.Drafts"
            , "From: moi@mon.hote\r\n"
            . "To: toi@ton.hote\r\n"
            . "Subject: test\r\n"
            . "\r\n"
            . "Ceci est un message de test. Ignorez le\r\n"
            );

$check = imap_check($stream);
echo "Nombre de messages après ajout : ". $check->Nmsgs . "\n";

imap_close($stream);
?>

```

8.54.11 `imap_base64()` : Décode un texte encodé en BASE64

string `imap_base64` (string text)

`imap_base64` décode un texte encodé en BASE64 (voir la [RFC2045](#) , Section 6.8). Le texte décodé est retourné sous la forme d'une chaîne

Voir aussi [imap_binary](#) , [base64_encode](#) et [base64_decode](#) .

8.54.12 `imap_binary()` : Convertit une chaîne à 8 bits en une chaîne à base64

string `imap_binary` (string string)

`imap_binary` convertit la chaîne à 8 bits string en une chaîne à base64 (selon la [RFC2045](#) , Section 6.8).

`imap_binary` retourne la chaîne codée.

Voir aussi [imap_base64](#) .

8.54.13 `imap_body()` : Lit le corps d'un message

string `imap_body` (resource imap_stream , int msg_number , int options)

`imap_body` retourne le corps du message numéro msg_number de la boîte aux lettres courante.

Le paramètre options est un masque qui peut contenir les valeurs suivantes :

- FT_UID - msg_number est un UID
- FT_PEEK - Ne pas lever le drapeau \Seen (Message lu) s'il n'est pas déjà levé.
- FT_INTERNAL - La chaîne renvoyée est au format interne, et ne va pas canoniser les CRLF.

`imap_body` va retourner une copie brute du corps du message. Pour extraire les sous-parties MIME du message, utilisez `imap_fetchstructure` pour analyser la structure, et `imap_fetchbody` pour extraire une copie d'une des sous-partie.

8.54.14 `imap_bodystruct()` : Lit la structure d'une section du corps d'un mail

objet `imap_bodystruct` (resource stream_id , int msg_no , string section)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.54.15 `imap_check()` : Vérifie le courrier de la boîte aux lettres courante

objet `imap_check` (resource imap_stream)

`imap_check` retourne les informations à propos de la boîte aux lettres courante. `imap_check` retourne FALSE en cas d'échec.

`imap_check` vérifie le statut de la boîte aux lettres courante, sur le serveur `imap_stream` , et retourne les informations dans un objet avec les membres suivants :

- Date - Date de dernière modification du contenu de la boîte aux lettres en accord avec la [RFC822](#)
- Driver - protocole utilisé pour accéder à la boîte aux lettres: POP3 , IMAP , NNTP .
- Mailbox - nom de la boîte aux lettres
- Nmsgs - nombre de messages de la boîte aux lettres
- Recent - nombre de messages récents de la boîte aux lettres

Exemple avec <code>imap_check</code>

```
<?php
$imap_obj = imap_check($imap_stream);
var_dump($imap_obj);
?>
```

L'exemple ci-dessus va afficher :

```
object(stdClass)(5) {
  ["Date"]=>
  string(37) "Wed, 10 Dec 2003 17:56:54 +0100 (CET)"
  ["Driver"]=>
  string(4) "imap"
  ["Mailbox"]=>
  string(54)
  "{www.example.com:143/imap/user="foo@example.com"}INBOX"
  ["Nmsgs"]=>
  int(1)
  ["Recent"]=>
  int(0)
```

8.54.16 `imap_clearflag_full()` : Supprime un flag (drapeau) sur un message

`bool imap_clearflag_full (resource stream , string sequence , string flag , string options)`

`imap_clearflag_full` efface le flag `flag` dans les messages de la séquence `sequence`, du flux `imap stream`. Les flags `flag` que vous pouvez effacer sont "`\\Seen`", "`\\Answered`", "`\\Flagged`", "`\\Deleted`" et "`\\Draft`" (tels que définis dans la RFC2060). Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec..

`options` est un masque de bit, qui accepte uniquement la valeur suivante :

- `ST_UID` - la séquence contient des UID au lieu de numéros de séquence.

Voir aussi `imap_setflag_full`.

8.54.17 `imap_close()` : Termine un flux IMAP

`bool imap_close (resource imap_stream , int flag)`

`imap_close` termine un flux IMAP. `imap_close` prend un argument optionnel `flag`, `CL_EXPUNGE`, qui va purger automatiquement la boîte aux lettres.

Voir aussi `imap_open`.

8.54.18 `imap_createmailbox()` : Crée une nouvelle boîte aux lettres

`bool imap_createmailbox (resource imap_stream , string mbox)`

`imap_createmailbox` crée une nouvelle boîte aux lettres nommée `mbox`. Les noms contenant des caractères spéciaux doivent être encodés.

`imap_createmailbox` retourne `TRUE` en cas de succès, et `FALSE` en cas d'erreur.

Exemple avec `imap_createmailbox`

```
<?php
$mbx = imap_open("{your.imap.host}", "utilisateur", "mot_de_passe", OP_HALFOPEN)
    or die("connexion impossible: " . imap_last_error());

$name1 = "nouvellepbox";
$name2 = imap_utf7_encode("nouvellepboxéx");

$newname = $name1;

echo "Le nouveau nom sera '$name1'<br />\n";

// Nous allons créer maintenant une nouvelle boîte aux lettres "phptestbox"
// dans votre dossier inbox, vérifier son état et, finalement, la supprimer
// pour remettre votre inbox dans son état initial.
if (@imap_createmailbox($mbx, imap_utf7_encode("{your.imap.host}INBOX.$newname"))) {
    $status = @imap_status($mbx, "{your.imap.host}INBOX.$newname", SA_ALL);
```

```

if ($status) {
    echo "Votre nouvelle boîte '$name1' est dans l'état suivant :<br />\n";
    echo "Messages : "      . $status->messages      . "<br />\n";
    echo "Récent : "       . $status->recent         . "<br />\n";
    echo "Non lus : "      . $status->unseen         . "<br />\n";
    echo "UID suivant : "  . $status->uidnext        . "<br />\n";
    echo "UID validité : " . $status->uidvalidity    . "<br />\n";

    if (imap_renamemailbox($mbox, "{your.imap.host}INBOX.$newname", "{your.imap.host}INBOX.$name2")) {
        echo "renommage de la boîte aux lettres '$name1' en '$name2'<br />\n";
        $newname = $name2;
    } else {
        echo "imap_renamemailbox sur la nouvelle boîte aux lettres a échoué : " . imap_last_error() . "<br />\n";
    }
} else {
    echo "imap_status sur la nouvelle boîte aux lettres a échoué : " . imap_last_error() . "<br />\n";
}

if(@imap_deletemailbox($mbox, "{your.imap.host}INBOX.$newname")) {
    echo "new mailbox supprimée pour remettre tout en état<br />\n";
} else {
    echo "imap_deletemailbox sur la nouvelle boîte aux lettres a échoué : " . implode("<br />\n", imap_errors()) . "<br />\n";
}

} else {
    echo "Impossible de créer une nouvelle boîte aux lettres : " . implode("<br />\n", imap_errors()) . "<br />\n";
}
imap_close($mbox);
?>

```

Voir aussi [imap_renamemailbox](#) , [imap_deletemailbox](#) et [imap_open](#) pour connaître le format des noms de mbox .

8.54.19 imap_delete() : Marque le fichier pour l'effacement, dans la boîte aux lettres courante

bool **imap_delete** (int imap_stream , int msg_number , int options)

[imap_delete](#) retourne TRUE .

[imap_delete](#) marque le fichier msg_number pour l'effacement, dans la boîte aux lettres courante. Le paramètre optionnel flags ne prend qu'une seule valeur, ft_uid , qui indique à PHP qu'il faut traiter msg_number comme un uid . L'effacement réel n'interviendra que lors de l'appel de la fonction [imap_expunge](#) ou de [imap_close](#) avec le paramètre optionnel CL_EXPUNGE.

Note

Les boîtes aux lettres POP3 n'ont pas les flags de leurs messages de sauvegardés entre les connexions, donc, la fonction [imap_expunge](#) doit être appelée pendant la même connexion pour que les messages marquées pour effacement soient réellement purgés.

Exemple avec [imap_delete](#)

```

<?php

$mbox = imap_open("{your.imap.host}INBOX", "utilisateur", "mot_de_passe")
    or die ("connexion impossible: " . imap_last_error());

$check = imap_mailboxmsginfo($mbox);
echo "Nombre de messages avant effacement : " . $check->Nmsgs . "<br />\n";

```

```

imap_delete($mbox, 1);

$check = imap_mailboxmsginfo ($mbox);
echo "Nombre de messages après effacement: " . $check->Nmsgs . "<br />\n" ;

imap_expunge ($mbox);

$check = imap_mailboxmsginfo ($mbox);
echo "Nombre de messages après imap_expunge: " . $check->Nmsgs . "<br />\n" ;

imap_close ($mbox);
?>

```

Voir aussi [imap_undelete](#) , [imap_expunge](#) et [imap_close](#) .

8.54.20 [imap_deletemailbox\(\)](#) : Efface une boîte aux lettres

bool [imap_deletemailbox](#) (resource [imap_stream](#) , string [mbox](#))

[imap_deletemailbox](#) efface la boîte aux lettres spécifiée. (voir [imap_open](#) pour connaître le format des noms de mbox).

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

Voir aussi [imap_createmailbox](#) , [imap_renamemailbox](#) et [imap_open](#) pour le format du paramètre mbox .

8.54.21 [imap_errors\(\)](#) : Retourne toutes les erreurs

array [imap_errors](#) (void)

[imap_errors](#) retourne tous les messages d'erreurs IMAP générés depuis le dernier appel à [imap_errors](#) , ou depuis le début de la page. Lorsque [imap_errors](#) est appelée, la pile d'erreur est vidée.

Voir aussi [imap_last_error](#) .

8.54.22 [imap_expunge\(\)](#) : Efface tous les messages marqués pour l'effacement

bool [imap_expunge](#) (resource [imap_stream](#))

[imap_expunge](#) efface tous les messages marqués pour l'effacement par [imap_delete](#) , [imap_mail_move](#) , ou [imap_setflag_full](#) .

[imap_expunge](#) retourne TRUE .

8.54.23 [imap_fetch_overview\(\)](#) : Lit un sommaire des en-têtes de messages

array [imap_fetch_overview](#) (resource [imap_stream](#) , string [sequence](#) , int [options](#))

imap_fetch_overview lit les en-têtes des courriers électroniques de la séquence `sequence` et retourne un sommaire de leur contenu. `sequence` va contenir une séquence d'indices de message ou d'UIDs, si `flags` contient `FT_UID`. La valeur retournée est un tableau d'objets : un par message d'en-tête décrit :

- `subject` - Le sujet du message
- `from` - Expéditeur
- `to` - Destinataire
- `date` - Date d'expédition
- `message_id` - Identification du message
- `references` - est une référence sur l'id de ce message
- `in_reply_to` - est une réponse à cet identifiant de message
- `size` - taille en octets
- `uid` - UID du message dans la boîte aux lettres
- `msgno` - numéro de séquence du message dans la boîte
- `recent` - Ce message est récent
- `flagged` - Ce message est marqué
- `answered` - Ce message a donné lieu à une réponse
- `deleted` - Ce message est marqué pour l'effacement
- `seen` - Ce message est déjà lu
- `draft` - Ce message est un brouillon

Exemple avec imap_fetch_overview

```
<?php
$mbx = imap_open("{your.imap.host:143}", "username", "password")
    or die("can't connect: " . imap_last_error());

$overview = imap_fetch_overview($mbx, "2,4:6", 0);

if (is_array($overview)) {
    foreach ($overview as $val) {
        echo "$val->msgno - $val->date - $val->subject\n";
    }
}

imap_close($mbx);
?>
```

8.54.24 imap_fetchbody() : Retourne une section extraite du corps d'un message

string **imap_fetchbody** (resource `imap_stream` , int `msg_number` , string `part_number` , int `options`)

imap_fetchbody va rechercher une section du corps du message, et la retourne sous la forme d'une chaîne. La section est une chaîne d'entiers, séparés par des virgules, qui servent d'index dans le corps du message, comme spécifié dans la norme IMAP4. Le texte n'est alors pas décodé par imap_fetchbody .

L'option imap_fetchbody est un masque qui peut contenir les valeurs suivantes :

- `FT_UID` - `msg_number` est un UID
- `FT_PEEK` - Ne pas lever le drapeau `\Seen` (Message lu) s'il n'est pas déjà levé.
- `FT_INTERNAL` - La chaîne renvoyée est au format interne, et ne va pas canoniser les CRLF.

Voir aussi [imap_fetchstructure](#) .

8.54.25 `imap_fetchheader()` : Retourne l'en-tête d'un message

string `imap_fetchheader` (resource `imap_stream` , int `msgno` , int `options`)

`imap_fetchheader` retourne l'en-tête brut et complet [RFC2822](#) du message `msgno` , sous la forme d'une chaîne.

Les options sont :

- FT_UID - L'argument `msgno` est un UID
- FT_INTERNAL - la chaîne renvoyée est au format "internal" , c'est-à-dire sans canonisation des CRLF
- FT_PREFETCHTEXT - RFC822.TEXT doit être pré téléchargé en même temps que l'en-tête. Cela réduit le RTT sur une connexion IMAP , si le message complet est souhaité. (e.g. dans une opération de sauvegarde dans un fichier).

8.54.26 `imap_fetchstructure()` : Lit la structure d'un message

object `imap_fetchstructure` (resource `imap_stream` , int `msg_number` , int `options`)

`imap_fetchstructure` lit la structure du message `msg_number` . `imap_fetchstructure` dispose d'un paramètre `options` , qui n'a qu'une seule valeur, `ft_uid` , pour indiquer que l'argument `msg_number` est un uid . `imap_fetchstructure` retourne un objet avec des propriétés d'enveloppe, de date interne, de taille, de structure de flags et de corps, ainsi qu'un objet pour chaque attachement. La structure est la suivante :

<code>type</code>	Type primaire de corps
<code>encoding</code>	Codage de transfert du corps
<code>ifsubtype</code>	TRUE s'il y a une chaîne de sous type
<code>subtype</code>	sous type MIME
<code>ifdescription</code>	TRUE s'il y a une chaîne de description
<code>description</code>	Chaîne de description du contenu
<code>ifid</code>	TRUE s'il y a une chaîne d'identification
<code>id</code>	Chaîne d'identification
<code>lines</code>	Nombre de lignes
<code>bytes</code>	Nombre d'octets
<code>ifdisposition</code>	TRUE s'il y a une chaîne de disposition
<code>disposition</code>	Chaîne de disposition
<code>ifdparameters</code>	TRUE s'il y a un tableau de paramètres <code>dparameters</code>
<code>dparameters</code>	tableau d'objets où chaque objet a une propriété "attribute" et une propriété "value" correspondant aux paramètres d'en-têtes Content-disposition MIME .
<code>ifparameters</code>	TRUE si le tableau de paramètres existe
<code>parameters</code>	Tableau d'objets où chacun a une propriété "attribute" et une propriété "value".
<code>parts</code>	Tableau d'objets décrivant chaque partie MIME du message

0	text
---	------

1	multipart
2	message
3	application
4	audio
5	image
6	vidéo
7	autre

0	7BIT
1	8BIT
2	BINARY
3	BASE64
4	QUOTED-PRINTABLE
5	OTHER

Voir aussi [imap_fetchbody](#) .

8.54.27 [imap_get_quota\(\)](#) : Lit les quotas des boîtes aux lettres

array [imap_get_quota](#) (resource [imap_stream](#) , string [quota_root](#))

[imap_get_quota](#) retourne un tableau contenant les valeurs de quota et courante de la boîte aux lettres [quota_root](#) . Le quota représente la taille maximale de votre boîte aux lettres. La valeur courante est l'espace actuellement utilisé par votre boîte aux lettres. [imap_get_quota](#) retournera FALSE en cas d'échec.

[imap_get_quota](#) ne fonctionne actuellement qu'avec les bibliothèques c-client2000.

NOTE : Pour que cette fonction fonctionne, la ressource de mail doit avoir été ouverte par un utilisateur administrateur du système de mail. Pour un utilisateur non administrateur du système de mail, référez-vous à la fonction PHP [imap_get_quotaroot](#) .

[imap_stream](#) doit avoir été créé avec la fonction [imap_open](#) . Ce flux est nécessairement ouvert en tant qu'administrateur du serveur, pour que les droits nécessaires lui soient alloués. [quota_root](#) doit être de la forme : " user.nom " , où "nom" est le nom de la boîte aux lettres que vous souhaitez analyser.

Exemple avec [imap_get_quota](#)

```
<?php
$mbx = imap_open("{votre.hote.imap}", "mailadmin", "mot de passe", OP_HALFOPEN)
    or die("Connexion impossible : ".imap_last_error());
$quota_value = imap_get_quota($mbx, "user.toto");
if(is_array($quota_value)) {
    echo "Utilisation actuelle : " . $quota_value['usage'];
    echo "Quota : " . $quota_value['limit'];
}
imap_close($mbx);
?>
```

Depuis PHP 4.3, la fonction reflète plus fidèlement les fonctionnalités édictée par la RFC 2087. Le tableau retourné a changé pour supporter un nombre illimité de ressources retournées (i.e. messages ou sous-dossiers) avec chaque ressource nommée qui est identifiée par une clé. Chaque clé contient alors un autre tableau avec l'utilisation et le quota. L'exemple ci-dessous

montre comment l'utiliser.

Pour des raisons de compatibilité, la méthode d'accès originale est toujours disponible, mais il est recommandé de l'abandonner.

Exemple avec `imap_get_quota` pour PHP 4.3

```
<?php
$mbx = imap_open("{your.imap.host}", "mailadmin", "password", OP_HALFOPEN)
    or die("Impossible de se connecter : ".imap_last_error());

$quota_values = imap_get_quota($mbx, "user.kalowsky");
if(is_array($quota_values)) {
    $storage = $quota_values['STORAGE'];
    echo "Utilisation actuelle de la capacité de stockage : " . $storage['usage'];
    echo "Quota actuel de stockage : " . $storage['limit'];

    $message = $quota_values['MESSAGE'];
    echo "Niveau d'utilisation de MESSAGE : " . $message['usage'];
    echo "Quota de MESSAGE : " . $message['limit'];

    /* ... */
}

imap_close($mbx);
?>
```

Voir aussi `imap_open` , `imap_set_quota` et `imap_get_quotaroot` .

8.54.28 `imap_get_quotaroot()` : Lit les quotas de chaque utilisateur

array `imap_get_quotaroot` (resource `imap_stream` , string `quota_root`)

`imap_get_quotaroot` retourne un tableau d'entiers, contenant les quotas de la boîte à lettres de l'utilisateur. Toutes les valeurs sont représentées par une clé basée sur le nom de la boîte, et par un tableau représentant le niveau d'utilisation et les limites.

La valeur limite représente l'espace maximal alloué à l'utilisateur. Le niveau d'utilisation représente l'occupation actuelle de la boîte. Cette fonction retournera FALSE si une erreur est survenue, et un tableau de données si la réponse du serveur n'a pu être comprise.

Cette fonction est uniquement accessible aux utilisateurs de la bibliothèque c-client2000 ou plus récent.

`imap_stream` est une ressource de connexion, obtenue grâce à `imap_open` . Cette connexion doit être ouverte avec l'identité de l'utilisateur que l'on étudie. `quota_root` doit être un nom de boîte aux lettres (i.e. INBOX).

Exemple avec `imap_get_quotaroot`

```
<?php
$mbx = imap_open("{your.imap.host}", "kalowsky", "password", OP_HALFOPEN)
    or die("can't connect: ".imap_last_error());

$quota = imap_get_quotaroot($mbx, "INBOX");
if(is_array($quota)) {
    $storage = $quota_values['STORAGE'];
    echo "STORAGE usage level is: " . $storage['usage'];
    echo "STORAGE limit level is: " . $storage['limit'];
}
```

```

$message = $quota_values['MESSAGE'];
echo "MESSAGE niveau d'utilisation : " . $message['usage'];
echo "MESSAGE niveau d'utilisation : " . $message['limit'];

/* ... */
}
imap_close($mbox);
?>

```

Voir aussi [imap_open](#) , [imap_set_quota](#) et [imap_get_quota](#) .

8.54.29 [imap_getacl\(\)](#) : Retourne le ACL pour la boîte aux lettres

array [imap_getacl](#) (resource stream_id , string mailbox)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

[imap_getacl](#) n'est actuellement disponible que pour les utilisateurs de la bibliothèque c-client2000 ou supérieur.

Voir aussi [imap_setacl](#) .

8.54.30 [imap_getmailboxes\(\)](#) : Liste les boîtes aux lettres, et retourne les détails de chacune

array [imap_getmailboxes](#) (resource imap_stream , string ref , string pattern)

[imap_getmailboxes](#) retourne un tableau d'objets contenant les informations sur les boîtes aux lettres. Chaque objet possède un attribut de name , qui contient le nom complet de la boîte aux lettres, delimiter qui est le délimiteur hiérarchique et attributes . attributes est un masque de bits, qui contient :

- LATT_NOINFERIORS - Cette boîte aux lettres n'a pas d'"enfants" (il n'y a plus de boîtes aux lettres en dessous de celle-ci).
- LATT_NOSELECT - Ceci est juste un container, pas une boîte aux lettres (vous ne pouvez pas l'ouvrir).
- LATT_MARKED - Cette boîte aux lettres est marquée. Utilisé uniquement avec UW-IMAPD.
- LATT_UNMARKED - Cette boîte aux lettres n'est pas marquée. Utilisé uniquement avec UW-IMAPD.

Les noms de boîtes aux lettres contenant des caractères internationaux hors de l'espace ASCII seront encodés et pourront être décodés avec [imap_utf7_decode](#) .

ref ne devrait être que le serveur sous la forme décrite dans [imap_open](#) , et pattern spécifie la position dans la hiérarchie des boîtes aux lettres, où il faut commencer à chercher. Si vous voulez passer en revue toute la hiérarchie, passez '*' comme pattern .

Il y a deux caractères spéciaux que vous pouvez utiliser dans pattern : '*' et '%'. '*' signifie : toutes les boîtes aux lettres. Si vous passez pattern comme '*', vous obtiendrez la liste complète des boîtes aux lettres de la hiérarchie. '%' signifie qu'on ne s'intéresse qu'au niveau courant. '%' passé à pattern ne retournera que les boîtes aux lettres de niveau supérieur; '~'/mail/%'. Sous UW_IMAPD

retournera toutes les boîtes aux lettres du dossier ~/mail directory , mais pas leurs enfants.

Exemple avec `imap_getmailboxes`

```
<?php
$mbx = imap_open("{your.imap.host}", "utilisateur", "mot_de_passe", OP_HALFOPEN)
    or die("connexion impossible : " . imap_last_error());

$list = imap_getmailboxes($mbx, "{votre.hote.imap}", "*");
if(is_array($list)) {
    reset($list);
    foreach ($list as $key => $val) {
        echo "($key) ";
        echo imap_utf7_decode($val->name).", ";
        echo "' ' . $val->delimiter . "' , ";
        echo $val->attributes . "<br />\n";
    }
} else {
    echo "imap_getmailboxes a échoué : " . imap_last_error() . "\n";
}

imap_close($mbx);
?>
```

Voir aussi [imap_getsubscribed](#) .

8.54.31 `imap_getsubscribed()` : Liste toutes les boîtes aux lettres souscrites

array `imap_getsubscribed` (resource `imap_stream` , string `ref` , string `pattern`)

`imap_getsubscribed` est identique à `imap_getmailboxes` , mais ne retourne que les boîtes aux lettres auxquelles l'utilisateur est inscrit.

8.54.32 `imap_header()` : Alias de `imap_headerinfo`

Cette fonction est un alias de : `imap_headerinfo` .

8.54.33 `imap_headerinfo()` : Lit l'en-tête du message

object `imap_headerinfo` (resource `imap_stream` , int `msg_number` , int `fromlength` , int `subjectlength` , string `defaulthost`)

`imap_headerinfo` retourne un objet contenant divers éléments d'en-tête.

Éléments retournés par `imap_headerinfo`

```
remail, date, Date, subject, Subject, in_reply_to, message_id,
newsgroups, followup_to, references
```

éléments d'en-tête :

```
Recent - 'R' si récent et lu
         'N' si récent et non lu
         ' ' si non récent
Unseen - 'U' si non lu ET non récent
         ' ' si lu OU non lu et récent
Answered - 'A' si répondu,
```

```

    ' ' si non répondu
Deleted - 'D' si effacé,
    ' ' si non effacé
Draft - 'X' si brouillon,
    ' ' si non brouillon
Flagged - 'F' si marqué,
    ' ' si non marqué

```

Notez bien que le comportement récent/non lu est un peu particulier : si vous voulez savoir si un message est non lu, vous devez le vérifier avec

```
Unseen == 'U' || Recent == 'N'
```

toaddress (toute la ligne d'en-tête To: jusqu'à 1024 caractères)

to[] (retourne un objet avec tout l'en-tête To, contenant):

```

personal
adl
mailbox
host

```

fromaddress (toute la ligne d'en-tête from: jusqu'à 1024 caractères)

from[] (retourne un objet avec tout l'en-tête From, contenant):

```

personal
adl
mailbox
host

```

ccaddress (toute la ligne d'en-tête CC: jusqu'à 1024 caractères)

cc[] (retourne un objet avec tout l'en-tête CC, contenant):

```

personal
adl
mailbox
host

```

bccaddress (toute la ligne d'en-tête BCC: jusqu'à 1024 caractères)

bcc[] (retourne un objet avec tout l'en-tête BCC, contenant):

```

personal
adl
mailbox
host

```

reply_toaddress (oute la ligne d'en-tête Reply_to: jusqu'à 1024 caractères)

reply_to[] (retourne un objet avec tout l'en-tête Reply_to, contenant)

```

personal
adl
mailbox
host

```

senderaddress (toute la ligne d'en-tête Sender: jusqu'à 1024 caractères)

sender[] (retourne un objet avec tout l'en-tête Sender, contenant)

```

personal
adl
mailbox
host

```

return_path (toute la ligne d'en-tête Return-path: jusqu'à 1024 caractères)

return_path[] (retourne un objet avec tout l'en-tête Return-path, contenant)

```

personal
adl
mailbox
host

```

update (Date du mail, au format UNIX)

```

fetchfrom (ligne d'en-tête from formatée pour tenir dans
fromlength
caractères)

fetchsubject (ligne d'en-tête subject formatée pour tenir dans
subjectlength
caractères)

```

8.54.34 `imap_headers()` : Retourne les en-têtes de tous les messages d'une boîte aux lettres

array `imap_headers` (resource imap_stream)

`imap_headers` retourne un tableau de chaînes contenant les en-têtes des messages : une chaîne par message.

8.54.35 `imap_last_error()` : Retourne la dernière erreur (si elle existe) qui est survenue lors de la dernière requête

string `imap_last_error` (void)

`imap_last_error` retourne le texte complet de la dernière erreur IMAP (si elle existe) qui est survenue lors de la dernière requête. La pile d'erreur n'est pas touchée. Appeler `imap_last_error` successivement, sans de nouvelles erreurs, retournera la même erreur.

Voir aussi [imap_errors](#) .

8.54.36 `imap_list()` : Lit la liste des boîtes aux lettres

array `imap_list` (resource imap_stream , string ref , string pattern)

`imap_list` retourne un tableau contenant la liste des noms des boîtes aux lettres. Voyez [imap_getmailboxes](#) pour une description des paramètres `ref` et `pattern` .

Exemple avec `imap_list`

```

<?php
$mbox = imap_open("{your.imap.host}", "username", "password", OP_HALFOPEN)
    or die("can't connect: " . imap_last_error());

$list = imap_list($mbox, "{your.imap.host}", "*");
if(is_array($list)) {
    reset($list);
    foreach ($list as $val) {
        echo imap_utf7_decode($val) . "<br />\n";
    }
} else {
    echo "imap_list a échoué : " . imap_last_error() . "\n";
}

imap_close($mbox);
?>

```

Voir aussi [imap_getmailboxes](#) .

8.54.37 `imap_listmailbox()` : Alias de `imap_list`

Cette fonction est un alias de : `imap_list` .

8.54.38 `imap_listscan()` : Lit la liste des boîtes aux lettres, et y recherche une chaîne

array `imap_listscan` (resource `imap_stream` , string `ref` , string `pattern` , string `content`)

`imap_listscan` retourne un tableau contenant les noms des boîtes aux lettres qui contiennent la chaîne `content` dans leur nom.

Cette fonction est similaire à `imap_listmailbox` , mais va aussi vérifier la présence de la chaîne `content` dans les messages de la boîte aux lettres.

Voyez aussi `imap_getmailboxes` pour une description des paramètres `ref` et `pattern` .

8.54.39 `imap_listsubscribed()` : Alias de `imap_lsub`

Cette fonction est un alias de : `imap_lsub` .

8.54.40 `imap_lsub()` : Liste toutes les boîtes aux lettres enregistrées

array `imap_lsub` (resource `imap_stream` , string `ref` , string `pattern`)

`imap_lsub` retourne un tableau de toutes les boîtes aux lettres que vous avez enregistré.

8.54.41 `imap_mail_compose()` : Crée un message MIME

string `imap_mail_compose` (array `envelope` , array `body`)

Exemple avec `imap_mail_compose`

```
<?php
$envelope["from"] = "joe@example.com";
$envelope["to"]   = "foo@example.com";
$envelope["cc"]   = "bar@example.com";

$part1["type"] = TYPEMULTIPART;
$part1["subtype"] = "mixed";

$filename = "/tmp/imap.c.gz";
$fp = fopen($filename, "r");
$contents = fread($fp, filesize($filename));
fclose($fp);

$part2["type"] = TYPEAPPLICATION;
$part2["encoding"] = ENCBINARY;
$part2["subtype"] = "octet-stream";
$part2["description"] = basename($filename);
$part2["contents.data"] = $contents;
```

```

$part3["type"] = TYPETEXT;
$part3["subtype"] = "plain";
$part3["description"] = "description3";
$part3["contents.data"] = "contents.data3\n\n\n\t";

$body[1] = $part1;
$body[2] = $part2;
$body[3] = $part3;

echo nl2br(imap_mail_compose($envelope, $body));

?>

```

8.54.42 `imap_mail_copy()` : Copie les messages spécifiés dans une boîte aux lettres

`bool imap_mail_copy (resource imap_stream , string msglist , string mbox , int options)`

`imap_mail_copy` copie les messages email spécifiés par `msglist` dans la boîte aux lettres nommée `mbox` . Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

`msglist` est un intervalle, et pas seulement une liste de numéros de message (comme décrit dans la [RFC2060](#)).

`options` est un masque, qui peut contenir une ou plusieurs des valeurs suivantes :

- `CP_UID` - la séquence de nombre contient des UIDS
- `CP_MOVE` - Efface les messages après copie.

Voir aussi [imap_mail_move](#) .

8.54.43 `imap_mail_move()` : Déplace des messages dans une boîte aux lettres

`bool imap_mail_move (resource imap_stream , string msglist , string mbox , int options)`

`imap_mail_move` déplace les messages spécifiés par `msglist` dans la boîte aux lettres `mbox` . Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec..

`msglist` est un intervalle, et pas seulement une liste de messages (comme décrit dans la [RFC2060](#)).

`options` est un champ de bit et peut contenir une seule valeur :

- `CP_UID` - La séquence de nombres contient des UID

Voir aussi [imap_mail_copy](#) .

8.54.44 `imap_mail()` : Envoie un message mail

`bool imap_mail (string to , string subject , string message , string additional_headers , string cc , string bcc , string rpath)`

`imap_mail` permet d'envoyer des mails avec une gestion correcte des destinataires Cc et Bcc.

Les paramètres `to` , `cc` et `bcc` sont tous des chaînes et sont analysées comme des listes d'adresses rfc822.

Les destinataires spécifiés dans `bcc` recevront le mail, mais seront absents des en-têtes.

Utilisez le paramètre `rpath` pour spécifier l'en-tête Return-Path. Cela est utile lorsque vous utilisez PHP comme un client mail pour plusieurs utilisateurs.

8.54.45 `imap_mailboxmsginfo()` : Lit les informations à propos de la boîte aux lettres courante

objet `imap_mailboxmsginfo` (resource `imap_stream`)

`imap_mailboxmsginfo` retourne les informations à propos de la boîte aux lettres courante.

`imap_mailboxmsginfo` retourne FALSE en cas d'échec.

`imap_mailboxmsginfo` vérifie le statut courant de la boîte aux lettres sur le serveur. C'est similaire à l'utilisation de la fonction `imap_status` , mais fournit également la taille totale des messages de la boîte aux lettres, ce qui demande un peu plus de temps à l'exécution. `imap_mailboxmsginfo` retourne un objet avec les propriétés suivantes :

Date	Date de dernière modification du contenu de la boîte aux lettres
Driver	Pilote
Mailbox	Nom de la boîte aux lettres
Nmsgs	Nombre de messages
Recent	Nombre de messages récents
Unread	Nombre de messages non lus
Deleted	Nombre de messages effacés
Size	Taille de la boîte aux lettres

Exemple avec `imap_mailboxmsginfo`

```
<?php
$mbx = imap_open("{your.imap.host}INBOX", "utilisateur", "mot_de_passe")
    or die("conexion impossible: ".imap_last_error());

$check = imap_mailboxmsginfo($mbx);

if($check) {
    echo "Date: "      . $check->Date      . "<br />\n" ;
    echo "Pilote: "   . $check->Driver   . "<br />\n" ;
    echo "Mailbox: " . $check->Mailbox . "<br />\n" ;
    echo "Messages: " . $check->Nmsgs . "<br />\n" ;
    echo "Récent: " . $check->Recent . "<br />\n" ;
    echo "Non lus: " . $check->Unread . "<br />\n" ;
    echo "Effacés: " . $check->Deleted . "<br />\n" ;
    echo "Taille: " . $check->Size . "<br />\n" ;
} else {
    echo "imap_check() a échoué: ".imap_last_error(). "<br />\n";
}

imap_close($mbx);
```

?>

8.54.46 `imap_mime_header_decode()` : Décode les éléments MIME d'un en-tête

array `imap_mime_header_decode` (string text)

`imap_mime_header_decode` décode un message MIME qui contient des données non ASCII (voir [RFC2047](#)). Les éléments décodés sont retournés dans un tableau d'objets. Chacun de ces objets a deux propriétés : "charset" et "text". Si l'élément n'a pas été encodé, ou, en d'autres termes, s'il est en clair (plain US_ASCII), la propriété "charset" est mise à "default".

Exemple `imap_mime_header_decode`

```
<?php
$text="=?ISO-8859-1?Q?Keld_J=F8rn_Simonsen?= <keld@dkuug.dk>";

$elements = imap_mime_header_decode($text);

for($i=0; $i<count($elements); $i++) {
    echo "Charset: {$elements[$i]->charset}\n";
    echo "Texte: {$elements[$i]->text}\n\n";
}
?>
```

Dans l'exemple ci-dessus, on trouve deux éléments : le premier a été encodé en ISO-8859-1, et le second est en clair.

8.54.47 `imap_msgno()` : Retourne le numéro de séquence de message pour un UID donné

int `imap_msgno` (resource imap_stream , int uid)

`imap_msgno` retourne le numéro de séquence de message pour l'UID uid . C'est la fonction inverse de `imap_uid` .

Voir aussi `imap_uid` .

8.54.48 `imap_num_msg()` : Retourne le nombre de message dans la boîte aux lettres courante

int `imap_num_msg` (resource imap_stream)

`imap_num_msg` retourne le nombre de message dans la boîte aux lettres courante.

Voir aussi `imap_num_recent` et `imap_status` .

8.54.49 `imap_num_recent()` : Retourne le nombre de messages récents dans la boîte aux lettres courante

int **imap_num_recent** (resource *imap_stream*)

imap_num_recent retourne le nombre de messages récents dans la boîte aux lettres courante.

Voir aussi imap_num_msg et imap_status .

8.54.50 **imap_open()** : Ouvre un flux IMAP vers une boîte aux lettres

resource **imap_open** (string *mailbox* , string *username* , string *password* , int *options*)

imap_open retourne un flux IMAP en cas de succès, et FALSE en cas d'erreur. imap_open peut aussi être utilisée pour ouvrir des flots sur des serveurs POP3 et NNTP .

Un nom de boîte aux lettres est constitué d'une adresse de serveur, et d'une adresse de boîte sur ce serveur. Le mot réservé INBOX représente la boîte aux lettres de l'utilisateur courant. L'adresse du serveur, mise entre accolades '{' et '}', est constituée du nom du serveur ou de son adresse IP, d'une spécification de protocole (commençant par '/') et d'un port optionnel (spécifié avec ':'). Cette partie est obligatoire dans les paramètres de la boîte aux lettres. Les noms de boîtes aux lettres qui contiennent des caractères spéciaux (en dehors de l'espace ASCII) doivent être encodés avec imap_utf7_encode .

Tous les noms commençant par { sont des noms distants et sont sous la forme "{
nom_systeme_distant [":" port] [flags] "}" [nom_mailbox] où :

- nom_systeme_distant - Nom de domaine Internet ou une adresse IP de serveur entouré de guillemets.
- port - numéro de port TCP (optionnel), la valeur par défaut est la valeur du port pour ce service.
- flags - flags optionnels, voir la table suivante.
- nom_mailbox - nom de la mailbox distante, par défaut : INBOX

Flag	Description
/service= service	service pour l'accès à la mailbox, par défaut : "imap"
/user= user	nom de l'utilisateur distant pour l'identification sur le serveur
/authuser= user	utilisateur distance d'identification ; si spécifié, ce sera le nom de l'utilisateur dont le mot de passe est utilisé (e.g. administratur)
/anonymous	accès distant en anonyme
/debug	la télémétrie d'enregistrement du protocole dans les logs de débogage de l'application
/secure	ne transmet pas un mot de passe en clair à travers le réseau
/imap , /imap2 , /imap2bis , /imap4 , /imap4rev1	équivalent de /service=imap
/pop3	équivalent de /service=pop3
/nntp	équivalent de /service=nntp
/norsh	ne pas utiliser rsh ou ssh pour établir une session de pré identification IMAP
/ssl	utilise Secure Socket Layer pour crypter la session
/validate-cert	valide les certificats depuis le serveur TLS/SSL (c'est le comportement par défaut)
/novalidate-cert	

8.54.49 **imap_num_recent()** : Retourne le nombre de messages récents dans la boîte aux lettres courante

	ne pas valider les certificats depuis le serveur TLS/SSL, nécessaire si le serveur utilise des certificats auto-signés
/tls	force l'utilisation de start-TLS pour crypter la session et rejète les connexions aux serveurs qui ne le supporte pas
/notls	n'utilise pas start-TLS pour crypter la session, y compris avec les serveurs qui le supporte
/readonly	demande un accès en lecture seule sur mailbox (IMAP uniquement ; ignoré sous NNTP, et une erreur avec SMTP et POP3)

options est un masque de bit, qui peut prendre une ou plusieurs des valeurs suivantes :

- OP_READONLY - Ouvre une boîte aux lettres en lecture seule
- OP_ANONYMOUS - Ne pas utiliser, ou modifier le fichier .newsrsrc pour les news.
- OP_HALFOPEN - Pour les noms IMAP et NNTP , ouvre une connexion mais n'ouvre pas une boîte aux lettres.
- CL_EXPUNGE - Supprime automatiquement la boîte aux lettres de la liste, lors de la terminaison du flux (voir aussi [imap_delete](#) and [imap_expunge](#))
- OP_DEBUG - négociations de débogage du protocole
- OP_SHORTCACHE - Cache court (elt uniquement)
- OP_SILENT - Ne pas transmettre les événements (utilisation interne)
- OP_PROTOTYPE - Retourne le prototype du driver
- OP_EXPUNGE - Nettoye silencieusement le flux recyclé
- OP_SECURE - Ne pas effectuer des identifications non sécuritées

Pour se connecter à un serveur IMAP , on peut utiliser la commande suivante :

Différentes utilisations de imap_open
<pre><?php // Pour se connecter à un serveur IMAP fonctionnant sur le port 143 de la // machine locale, faites ceci : \$mailbox = imap_open("{localhost:143}INBOX","user_id","password"); // Pour se connecter à un serveur POP3 fonctionnant sur le port 110 du // serveur local, faites ceci : \$mailbox = imap_open("{localhost:110/pop3}INBOX","user_id","password"); // Pour se connecter à un serveur SSL IMAP ou POP3, ajoutez /ssl // après la spécification du protocole \$mailbox = imap_open("{localhost:993/imap/ssl}INBOX", "user_id", "password"); // Pour se connecter à un serveur SSL IMAP ou POP3 avec un certificat auto-signé // ajoutez /ssl/novalidate-cert après le protocole : \$mailbox = imap_open("{localhost:995/pop3/ssl/novalidate-cert}", "user_id", "password"); // Pour se connecter à un serveur NNTP qui fonctionne sur // le port 119 de la machine locale on peut utiliser la commande: \$nnntp = imap_open("{localhost:119/nnntp}comp.test","",""); // Pour se connecter à un serveur distant, remplacez "localhost" par // le nom ou l'adresse IP de la machine. ?></pre>

Exemple avec imap_open
<pre><?php \$mailbox = imap_open("{votre.hote.imap:143}", "nom_utilisateur", "mot de passe"); echo "<h1>Mailboxes</h1>\n"; \$folders = imap_listmailbox(\$mailbox, "{votre.hote.imap:143}", "*");</pre>

```

if ($folders == false) {
    echo "Appel échoué<br />\n";
} else {
    foreach ($folders as $val) {
        echo $val . "<br />\n";
    }
}

echo "<h1>en-têtes dans INBOX</h1>\n";
$headers = imap_headers($mbox);
if ($headers == false) {
    echo "Appel échoué<br />\n";
} else {
    foreach ($headers as $val) {
        echo $val . "<br />\n";
    }
}

imap_close($mbox);
?>

```

8.54.51 `imap_ping()` : Vérifie que le flux IMAP est toujours actif

bool `imap_ping` (resource `imap_stream`)

`imap_ping` retourne TRUE si le flux `imap_stream` existe toujours, et FALSE sinon.

`imap_ping` vérifie que le flux IMAP est toujours actif, en lui envoyant un ping. Cette fonction permet de se rendre compte qu'un mail est arrivé : c'est même la méthode préconisée pour des tests périodiques de vérification du courrier. Cette fonction peut aussi servir à garder une connexion ouverte, avec les serveurs dotés d'un délai d'expiration.

Exemple avec `imap_ping`

```

<?php

$imap = imap_open("{votre.hote.imap}", "mailadmin", "motdepasse");

// après une pause
if (!imap_ping($imap)) {
    // effectuez un traitement pour se reconnecter
}

?>

```

8.54.52 `imap_qprint()` : Convertit une chaîne à guillemets en une chaîne à 8 bits

string `imap_qprint` (string `string`)

`imap_qprint` convertit la chaîne à guillemets `string` en une chaîne à 8 bits (selon la [RFC2045](#), section 6.7).

Voir aussi [imap_8bit](#) .

8.54.53 `imap_renamemailbox()` : Renomme une boîte aux lettres

bool `imap_renamemailbox` (resource `imap_stream` , string `old_mbox` , string `new_mbox`)

`imap_renamemailbox` renomme la boîte aux lettres `old_mbox` en `new_mbox` .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `imap_createmailbox` , `imap_deletemailbox` et `imap_open` pour le format de mbox .

8.54.54 `imap_reopen()` : Réouvre un flux IMAP vers une nouvelle boîte aux lettres

bool `imap_reopen` (resource `imap_stream` , string `mailbox` , string `options`)

`imap_reopen` réouvre la connexion spécifiée au serveur IMAP ou NNTP , avec une nouvelle boîtes aux lettres.

`options` est un masque de bit, qui peut contenir les valeurs suivantes :

- `OP_READONLY` - Ouvre une boîte aux lettres en lecture seule
- `OP_ANONYMOUS` - Ne pas utiliser, ou modifier le fichier `.newsrsrc` pour les news
- `OP_HALFOPEN` - Pour les noms IMAP et NNTP , ouvre une connexion mais n'ouvre pas une boîte aux lettres.
- `CL_EXPUNGE` - Supprime automatiquement la boîte aux lettres de la liste, lors de la terminaison du flux. (voir `imap_delete` et `imap_expunge`).

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.54.55 `imap_rfc822_parse_adrlist()` : Analyse une adresse email

array `imap_rfc822_parse_adrlist` (string `address` , string `default_host`)

`imap_rfc822_parse_adrlist` analyse la chaîne `address` , tel que définie dans la [RFC2822](#) et essaie, pour chaque adresse, de retourner un tableau d'objets. Les 4 membres de ces objets sont :

- `mailbox` - Le nom de la boîte aux lettres (nom d'utilisateur)
- `host` - Le nom de l'hôte
- `personal` - Le nom personnel
- `adl` - at domain source route (NDT : ???).

Exemple avec `imap_rfc822_parse_adrlist`

```
<?php
$address_string = "Hartmut Holzgraefe <hartmut@cvs.php.net>, postmaster@somedomain.net, root";
$address_array = imap_rfc822_parse_adrlist($address_string, "somedomain.net");

if (!is_array($address_array) || count($address_array) < 1) {
    die("Erreur!\n");
}

reset($address_array);
while(list($key, $val) = each($address_array)) {
    echo "boîte : " . $val->mailbox . "<br />\n";
}
```

```

echo "hôte   : " . $val->host . "<br />\n";
echo "nom    : " . $val->personal . "<br />\n";
echo "adl    : " . $val->adl . "<br />\n";
}
?>

```

8.54.56 `imap_rfc822_parse_headers()` : Analyse un en-tête mail

objet `imap_rfc822_parse_headers` (string headers , string defaulthost)

`imap_rfc822_parse_headers` analyse la chaîne headers et retourne un objet contenant différents éléments, similaires à la fonction `imap_header` , hormis les flags et autres éléments liés au serveur IMAP .

8.54.57 `imap_rfc822_write_address()` : Retourne une adresse email formatée correctement

string `imap_rfc822_write_address` (string mailbox , string host , string personal)

`imap_rfc822_write_address` retourne une adresse email formatée correctement selon la [RFC2822](#) , à partir du nom de la boîte aux lettres mailbox de l'hôte host , et des informations personnelles personal .

Exemple avec `imap_rfc822_write_address`

```

<?php
echo imap_rfc822_write_address("hartmut", "cvs.php.net", "Hartmut Holzgraefe")."\n";
?>

```

8.54.58 `imap_scanmailbox()` : Alias de `imap_listscan`

Cette fonction est un alias de : `imap_listscan` .

8.54.59 `imap_search()` : Retourne un tableau de message après recherche

array `imap_search` (resource imap_stream , string criteria , int options , string charset)

`imap_search` effectue une recherche dans la boîte aux lettres courante, sur le flux IMAP courant. criteria est une chaîne, délimitée par des espaces, dans laquelle les mots-clés suivants sont acceptés. Tous les arguments multi-mots doivent être entre guillemets :

- ALL - retourne tous les messages qui vérifient le reste du critère.
- ANSWERED - tous les messages avec le flag \ANSWERED
- BCC "string" - tous les messages avec la chaîne "string" dans le champ Bcc :
- BEFORE "date" - tous les messages avec Date : avant "date"
- BODY "string" - tous les messages avec "string" dans le corps
- CC "string" - tous les messages avec "string" dans le champ Cc :
- DELETED - tous les messages effacés
- FLAGGED - tous les messages avec le flag \FLAGGED (parfois interprété comme Important ou Urgent)

- FROM "string" - tous les messages avec la chaîne "string" dans le champ From :
- KEYWORD "string" - tous les messages avec la chaîne "string" comme mot-clé
- NEW - tous les nouveaux messages
- OLD - tous les anciens messages
- ON "date" - tous les messages avec la date "date" comme champ Date :
- RECENT - tous les messages avec le flag \RECENT
- SEEN - tous les messages lus (avec le flag \SEEN flag)
- SINCE "date" - tous les messages avec la date Date: après "date"
- SUBJECT "string" - tous les messages avec la chaîne "string" dans le champ Subject :
- TEXT "string" - tous les messages avec le texte "string"
- TO "string" - tous les messages avec la chaîne "string" dans le champ To :
- UNANSWERED - tous les messages non répondus
- UNDELETED - tous les messages non effacés
- UNFLAGGED - tous les messages non flaggés
- UNKEYWORD "string" - tous les messages ne contenant pas le mot-clé "string"
- UNSEEN - tous les messages non lus

Par exemple, pour rechercher les messages non répondus, envoyés par maman, vous pouvez utiliser : "UNANSWERED FROM maman". Les recherches semblent insensibles à la casse. Cette liste de critères est issue du code d'un client C UW et peut être incomplète ou imprécise. (voir aussi RFC2060, section 6.4.4).

Les valeurs pour les flags sont SE_UID, qui fait que le tableau réponse contient les UIDs plutôt que les numéros de séquence.

Le paramètre charset a été ajouté en PHP 4.3.3.

8.54.60 `imap_set_quota()` : Modifie le quota d'une boîte aux lettres

bool `imap_set_quota` (resource `imap_stream` , string `quota_root` , int `quota_limit`)

`imap_set_quota` modifie le quota de la boîte aux lettres `quota_root` , en la fixant à `quota_limit` . `imap_set_quota` requiert que `imap_stream` ait été ouvert avec un compte d'administrateur, pour avoir les droits nécessaires : elle ne fonctionnera avec aucun autre utilisateur.

`imap_get_quota` ne fonctionne actuellement qu'avec les bibliothèques c-client2000.

`imap_stream` doit avoir été créé avec la fonction `imap_open` . Ce flux est nécessairement ouvert en tant qu'administrateur du serveur, pour que les droits nécessaires lui soit alloués. `quota_root` doit être de la forme : " user.nom " , où "nom" est le nom de la boîte aux lettres que vous souhaitez analyser. `quota_limit` est la nouvelle taille maximum (en ko) de la boîte `quota_root` .

`imap_set_quota` retourne TRUE en cas de succès, et FALSE sinon.

Exemple avec `imap_set_quota`

```
<?php
$mbx = imap_open("{your.imap.host:143}", "mailadmin", "password");

if (!imap_set_quota($mbx, "user.kalowsky", 3000)) {
    echo "Error in setting quota\n";
    return;
}

imap_close($mbx);
?>
```

Voir aussi [imap_open](#) et [imap_get_quota](#) .

8.54.61 [imap_setacl\(\)](#) : Modifie le ACL de la boîte aux lettres

bool **imap_setacl** (resource stream_id , string mailbox , string id , string rights)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

[imap_setacl](#) n'est actuellement disponible que pour les utilisateurs de la bibliothèque c-client2000 ou supérieur.

Voir aussi [imap_getacl](#) .

8.54.62 [imap_setflag_full\(\)](#) : Positionne un flag sur un message

bool **imap_setflag_full** (resource stream , string sequence , string flag , *string options*)

[imap_setflag_full](#) affecte le flag spécifié aux messages de la séquence donnée.

Les flags que vous pouvez modifier sont "\\Seen", "\\Answered", "\\Flagged", "\\Deleted", "\\Draft" et "\\Recent" (comme défini dans la RFC2060).

options est un masque de bit, qui accepte uniquement la valeur suivante :

- ST_UID - la séquence contient des UID au lieu de numéros de séquence.

Exemple avec [imap_setflag_full](#)

```
<?php
$mbx = imap_open("{your.imap.host:143}", "username", "password")
    or die("can't connect: " . imap_last_error());

$status = imap_setflag_full($mbx, "2,5", "\\Seen \\Flagged");

echo gettype($status) . "\n";
echo $status . "\n";

imap_close($mbx);
?>
```

Voir aussi [imap_clearflag_full](#) .

8.54.63 [imap_sort\(\)](#) : Trie des messages

array **imap_sort** (resource stream , int criteria , int reverse , *int options* , *string search_criteria* , *string charset*)

[imap_sort](#) retourne un tableau de numéros de messages, triés en fonction des paramètres suivants :

reverse vaut 1 pour signifier : tri inverse.

Les critères criteria peuvent être un (et un seul) parmi les suivants :

- SORTDATE - Date du message
- SORTARRIVAL - Date d'arrivée
- SORTFROM - Nom de la première boîte aux lettres de l'adresse d'origine (From address)
- SORTSUBJECT - Sujet du message
- SORTTO - Nom de la première boîte aux lettres de destination (To address)
- SORTCC - Nom de la boîte aux lettres de copie cachée (cc address)
- SORTSIZE - Taille du message en octets

Les flags dont des masques de bits, d'un ou plusieurs des éléments suivants :

- SE_UID - Retourne des UID à la place de numéros
- SE_NOPREFETCH - Ne pas pré-télécharger les messages trouvés

Le paramètre charset a été ajouté en PHP 4.3.3.

8.54.64 `imap_status()` : Retourne les informations de statut sur une boîte aux lettres autres que la boîte courante

object `imap_status` (resource `imap_stream` , string `mailbox` , int `options`)

`imap_status` retourne un objet contenant les informations de statut. Les options valables sont :

- SA_MESSAGES - met la valeur de `status->messages` au nombre de messages dans la boîte aux lettres.
- SA_RECENT - met la valeur de `status->recent` au nombre de messages récents dans la boîte aux lettres.
- SA_UNSEEN - met la valeur de `status->unseen` au nombre de messages non lus dans la boîte aux lettres.
- SA_UIDNEXT - met la valeur de `status->uidnext` à la prochaine valeur d'uid qui sera utilisée.
- SA_UIDVALIDITY - met la valeur de `status->uidvalidity` à une constante, qui change lorsque l'uid de la boîte aux lettres n'est plus valide.
- SA_ALL - fixe toutes les valeurs précédentes.

`status->flags` est aussi fixé : c'est un masque de bit qui peut contenir tous les flags ci-dessus.

Exemple avec `imap_status`

```
<?php
$mbx = imap_open("{your.imap.host}", "utilisateur", "mot_de_passe", OP_HALFOPEN)
    or die("can't connect: " . imap_last_error());

$status = imap_status($mbx, "{your.imap.host}INBOX", SA_ALL);
if($status) {
    echo "Messages:      " . $status->messages      . "<br />\n";
    echo "Récents:      " . $status->recent        . "<br />\n";
    echo "Non lus:      " . $status->unseen         . "<br />\n";
    echo "UIDnext:     " . $status->uidnext         . "<br />\n";
    echo "UIDvalidité:  " . $status->uidvalidity    . "<br />\n";
} else {
    echo "imap_status a échoué : " . imap_last_error() . "\n";
}

imap_close($mbx);
?>
```

8.54.65 `imap_subscribe()` : Souscrit à une boîte aux lettres

bool `imap_subscribe` (resource `imap_stream` , string `mbox`)

`imap_subscribe` souscrit à la boîte aux lettres `mbox` .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `imap_unsubscribe` .

8.54.66 `imap_thread()` : Retourne l'arbre des messages organisés par thread

array `imap_thread` (resource `stream_id` , int `options`)

`imap_thread` retourne un tableau associatif contenant un arbre de messages organisés par thread par REFERENCES ou FALSE en cas d'erreur.

Chaque message dans la boîte aux lettres courante sera représenté par des entrées sous forme d'arbre dans le tableau résultant :

- `$thread["XX.num"]` - numéro du message courant
- `$thread["XX.next"]`
- `$thread["XX.branch"]`

Exemple avec `imap_thread`

```
<?php
// Ici, nous affichons les threads d'un newsgroup, en HTML

$nntp = imap_open('{news.example.com:119/nntp}un.newsgroup', '', '');
$threads = imap_thread($nntp);

foreach ($threads as $key => $val) {
    $tree = explode('.', $key);
    if ($tree[1] == 'num') {
        $header = imap_headerinfo($nntp, $val);
        echo "<ul>\n\t<li>" . $header->fromaddress . "\n";
    } elseif ($tree[1] == 'branch') {
        echo "\t</li>\n</ul>\n";
    }
}

imap_close($nntp);
?>
```

8.54.67 `imap_timeout()` : Configure ou retourne le timeout

mixed `imap_timeout` (int `timeout_type` , int `timeout`)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.54.68 `imap_uid()` : Retourne l'UID d'un message

int `imap_uid` (resource `imap_stream` , int `msgno`)

`imap_uid` retourne l'UID pour le message `msgno` . Un UID est un identifiant unique que ne change jamais, alors que le numéro du message dans la liste des messages peut changer à toute modification de la boîte aux lettres. C'est la fonction inverse de `imap_msgno` .

Note
Cette fonctionnalité n'est pas supportée par les boîtes aux lettres POP3. Voir aussi <code>imap_msgno</code> .

8.54.69 `imap_undelete()` : Enlève la marque d'effacement d'un message

bool `imap_undelete` (resource `imap_stream` , int `msg_number` , int `flags`)

`imap_undelete` enlève la marque d'effacement du message `msg_number` , placée avec `imap_delete` ou `imap_mail_move` .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `imap_delete` et `imap_mail_move` .

8.54.70 `imap_unsubscribe()` : Termine la souscription à une boîte aux lettres

bool `imap_unsubscribe` (string `imap_stream` , string `mbox`)

`imap_unsubscribe` termine la souscription à la boîte aux lettres `mbox` .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `imap_subscribe` .

8.54.71 `imap_utf7_decode()` : Décode une chaîne encodée en UTF-7 modifié

string `imap_utf7_decode` (string `text`)

`imap_utf7_decode` décode la chaîne UTF-7 modifié `text` en ISO-8859-1.

`imap_utf7_decode` retourne les données décodées en ISO-8859-1, ou FALSE si la chaîne `text` n'est pas au format UTF-7 modifié.

Cette fonction sert à encoder les noms de boîtes aux lettres qui contiennent des caractères internationaux hors de l'espace ASCII.

L'encodage UTF-7 modifié est défini dans la [RFC 2060](#) , section 5.1.3 (l'encodage UTF-7 original est lui défini dans [RFC1642](#)).

Voir aussi [imap_utf7_encode](#) .

8.54.72 imap_utf7_encode() : Convertit une chaîne ISO-8859-1 en texte UTF-7 modifié

string **imap_utf7_encode** (string data)

[imap_utf7_encode](#) convertit les données data en UTF-7 modifié. Notez que data doit être encodée en ISO-8859-1.

Cette fonction sert à encoder les noms de boîtes aux lettres qui contiennent des caractères internationaux hors de l'espace ASCII.

L'encodage UTF-7 modifié est défini dans la [RFC 2060](#) , section 5.1.3 (l'encodage UTF-7 original est lui défini dans [RFC1642](#)).

Voir aussi [imap_utf7_decode](#) .

8.54.73 imap_utf8() : Convertit du texte au format MIME en UTF8

string **imap_utf8** (string mime_encoded_text)

[imap_utf8](#) convertit le texte mime_encoded_text en UTF8. Les spécifications de MIME et UTF8 sont décrites dans les [RFC2047](#) et [RFC2044](#) .

8.55 Options PHP et informations

8.55.1 Introduction

Ces fonctions vous donnent accès à de nombreuses informations sur PHP lui-même, comme les configurations d'exécution, les extensions chargées, les versions, etc. Vous trouverez aussi des fonctions pour modifier des options. Ainsi que la star des fonctions PHP [phpinfo](#) .

8.55.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.55.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.55.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable	Historique
<code>assert.active</code>	"1"	PHP_INI_ALL	
<code>assert.bail</code>	"0"	PHP_INI_ALL	
<code>assert.warning</code>	"1"	PHP_INI_ALL	
<code>assert.callback</code>	NULL	PHP_INI_ALL	
<code>assert.quiet_eval</code>	"0"	PHP_INI_ALL	
<code>enable_dl</code>	"1"	PHP_INI_SYSTEM	
<code>max_execution_time</code>	"30"	PHP_INI_ALL	
<code>max_input_time</code>	"-1"	PHP_INI_PERDIR	Disponible depuis PHP 4.3.0.
<code>magic_quotes_gpc</code>	"1"	PHP_INI_PERDIR	Disponible depuis PHP <= 4.2.3.
<code>magic_quotes_runtime</code>	"0"	PHP_INI_ALL	

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

`assert.active` boolean

Active les évaluations de type `assert` .

`assert.bail` boolean

Termine le script si une assertion échoue.

`assert.warning` boolean

Emet une alerte PHP pour chaque assertion échouée.

`assert.callback` string

Fonction définie par le programmeur, à appeler pour chaque assertion échouée.

`assert.quiet_eval` boolean

Utilise la configuration courante de `error_reporting` durant les évaluations d'assertions. Si activée, aucune erreur n'est affichée (`error_reporting(0)` implicite)

durant l'évaluation. Si désactivée, les erreurs sont affichées en fonction de la configuration de error_reporting

enable_dl boolean

Cette directive est réellement utile lorsque PHP est compilé comme module Apache. Vous pouvez activer le chargement dynamique d'extension avec la fonction PHP dl au cas par cas, pour chaque serveur virtuel.

La raison principale pour désactiver ce système est la sécurité. Avec le chargement dynamique, il est possible de passer outre les configurations de safe_mode et open_basedir.

Par défaut, le chargement dynamique est autorisé, sauf avec le safe_mode. En safe_mode, il est toujours impossible d'utiliser la fonction dl.

max_execution_time entier

Fixe le temps maximal d'exécution d'un script, en secondes. Cela permet d'éviter que des scripts en boucles infinies saturent le serveur. La configuration par défaut est de 30 secondes.

Le temps d'exécution maximum n'est pas affecté par des appels systèmes tels que sleep. Reportez-vous à la fonction set_time_limit pour plus de détails.

Vous ne pouvez pas modifier la valeur de cette directive avec ini_set lorsque PHP est configuré en safe_mode. Le seul moyen de le faire est de désactiver le safe_mode ou de changer la valeur dans php.ini.

Votre serveur web peut avoir d'autres temps d'exécution maximum. E.g. Apache a une directive Timeout, IIS a une fonction de temps d'exécution maximum pour les CGI, les deux de 300 secondes par défaut. Lisez la documentation de votre serveur web la signification de cela.

max_input_time entier

Cette option spécifie la durée maximale pour recevoir les données d'entrée, via POST, GET et téléchargement de fichier.

magic_quotes_gpc boolean

Fixe le mode magic_quotes pour les opérations GPC (Get/Post/Cookie). Lorsque magic_quotes est activé, tous les caractères ' (guillemets simples), " (guillemets doubles), \ (antislash) et NUL sont échappés avec un antislash.

Note

Si la directive magic_quotes_sybase est aussi activée, elle écrasera magic_quotes_gpc. Avec les deux directives activées, seuls les guillemets simples seront protégés avec un autre guillemet simple. Les guillemets doubles, les antislashes et les NUL ne seront pas protégés.

Voir aussi get_magic_quotes_gpc.

magic_quotes_runtime boolean

Si magic_quotes_runtime est activé, toutes les fonctions qui obtiennent des données auprès d'une source externe, y compris les bases de données et les fichiers texte, verront leur guillemets échappés avec un antislash. Si magic_quotes_sybase est aussi activé, les guillemets simples seront échappés avec un autre guillemet simple, plutôt qu'un antislash.

8.55.5 Types de ressources

Cette extension ne définit aucune ressource.

8.55.6 Constantes pré-définies

Les constantes listées ici sont toujours disponibles dans PHP.

Constante	Valeur	Description
CREDITS_GROUP	1	Une liste des développeurs principaux
CREDITS_GENERAL	2	Crédits généraux. Design du langage, concepts, auteurs de PHP et module SAPI.
CREDITS_SAPI	4	Une liste des API de serveurs, et leurs auteurs.
CREDITS_MODULES	8	Une liste des extensions de PHP, et leurs auteurs
CREDITS_DOCS	16	Les crédits de l'équipe de documentation
CREDITS_FULLPAGE	32	Généralement utilisé combiné avec d'autres options. Cette option indique qu'une page HTML complète doit être générée.
CREDITS_QA	64	Les crédits pour le groupe d'assurance qualité.
CREDITS_ALL	-1	Tous les crédits. C'est l'équivalent de :CREDITS_DOCS + CREDITS_GENERAL + CREDITS_GROUP + CREDITS_MODULES + CREDITS_QA CREDITS_FULLPAGE. Elle génère une page HTML complète et autonome. C'est la valeur par défaut.
Constante	Valeur	Description
INFO_GENERAL	1	La ligne de configuration, le chemin du php.ini , la date de compilation, le système et plus encore.
INFO_CREDITS	2	Crédits de PHP. Voir aussi phpcredits .
INFO_CONFIGURATION	4	Valeurs locales et serveurs des directives PHP. Voir aussi ini_get .
INFO_MODULES	8	Les modules chargés et leurs configurations respectives.
INFO_ENVIRONMENT	16	Les variables d'environnement, qui sont aussi disponibles dans <code>\$ENV</code> .
INFO_VARIABLES	32	Toutes les <u>variables pré-définies</u> : EGPCS (Environnement, GET, POST, Cookie, Server).
INFO_LICENSE	64	La licence PHP. Voir aussi la FAQ de la licence .
INFO_ALL	-1	Affiche toutes les valeurs citées ci-dessus. C'est la valeur par défaut.

ASSERT_ACTIVE ([entier](#))
 ASSERT_CALLBACK ([entier](#))
 ASSERT_BAIL ([entier](#))
 ASSERT_WARNING ([entier](#))
 ASSERT_QUIET_EVAL ([entier](#))

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [assert_options](#)
- [assert](#)

- [dl](#)
- [extension_loaded](#)
- [get_cfg_var](#)
- [get_current_user](#)
- [get_defined_constants](#)
- [get_extension_funcs](#)
- [get_include_path](#)
- [get_included_files](#)
- [get_loaded_extensions](#)
- [get_magic_quotes_gpc](#)
- [get_magic_quotes_runtime](#)
- [get_required_files](#)
- [getenv](#)
- [getlastmod](#)
- [getmygid](#)
- [getmyinode](#)
- [getmypid](#)
- [getmyuid](#)
- [getopt](#)
- [getrusage](#)
- [ini_alter](#)
- [ini_get_all](#)
- [ini_get](#)
- [ini_restore](#)
- [ini_set](#)
- [main](#)
- [memory_get_usage](#)
- [php_ini_scanned_files](#)
- [php_logo_guid](#)
- [php_sapi_name](#)
- [php_uname](#)
- [phpcredits](#)
- [phpinfo](#)
- [phpversion](#)
- [putenv](#)
- [restore_include_path](#)
- [set_include_path](#)
- [set_magic_quotes_runtime](#)
- [set_time_limit](#)
- [version_compare](#)
- [zend_logo_guid](#)
- [zend_version](#)

8.55.8 **assert()** : Vérifie si une assertion est fausse

bool **assert** (mixed assertion)

assert va vérifier l'assertion assertion et prendre la mesure appropriée si le résultat est FALSE .

Si assertion est donnée sous la forme d'une chaîne, elle sera évaluée comme un code PHP par la fonction assert . Les avantages de ce type d'assertion sont d'être moins lourd si la vérification d'assertion est désactivée, et le contenu des messages lorsque l'assertion échoue.

Il est recommandé de n'utiliser les assertions que comme outil de débogage. Vous pouvez les utiliser pour les vérifications d'usage : ces conditions doivent normalement être vraies, et indiquer une erreur de programmation si ce n'est pas le cas. Vous pouvez aussi vérifier la présence de certaines extensions ou limitations du système.

Les assertions ne doivent pas être utilisées pour faire des opérations de vérifications en production, comme des vérifications de valeur d'argument. En conditions normales, votre code doit être en état de fonctionner si la vérification d'assertion est désactivée.

Le comportement de `assert` peut être configuré par `assert_options` ou par les configurations `.ini` décrites dans la page de manuel de cette fonction.

La fonction `assert_options` et la directive `ASSERT_CALLBACK` permettent de configurer une fonction qui sera appelée lorsque l'assertion échoue.

Les fonctions de callback pour `assert` sont particulièrement utiles pour bâtir des suites de tests automatiques, car elles vous permettent de capturer facilement le code passé à l'assertion, ainsi que des informations sur le lieu et le moment de l'assertion. Même si ces informations peuvent être appelées par d'autres méthodes, les assertions sont plus rapides et plus faciles.

La fonction de callback doit accepter trois arguments. Le premier contient le nom du fichier qui a vu l'assertion échouer. Le second contient le numéro de ligne dans le fichier précédent. Le troisième argument contient l'expression qui a échoué (s'il y en a : les valeurs littérales comme 1 ou "deux" ne seront pas passées par cet argument).

Gestion des assertions avec un gestionnaire personnalisé
<pre> <?php // Activation des assertions et mise en mode discret. assert_options(ASSERT_ACTIVE, 1); assert_options(ASSERT_WARNING, 0); assert_options(ASSERT_QUIET_EVAL, 1); // Création d'un gestionnaire d'assertions function mon_assert_handler(\$file, \$line, \$code) { echo "<hr />Assertion Failed: File '\$file'
 Line '\$line'
 Code '\$code'
<hr />"; } // Configuration de la méthode de callback assert_options (ASSERT_CALLBACK, 'mon_assert_handler'); // Utilisation d'une assertion qui va échouer assert ('mysql_query("")'); ?> </pre>

8.55.9 dl() : Charge une extension PHP à la volée

int **dl** (string library)

`dl` charge l'extension PHP `library` à la volée. Le paramètre `library` est **seulement** le nom de fichier de l'extension, qui dépend de votre plate-forme. Par exemple l'extension `sockets` (si compilée comme module partagé, et non par défaut), sera appelée `sockets.so` sous Unix, et `php_sockets.dll` sous Windows.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.. Si la fonctionnalité de chargement de module n'est pas disponible (voir Note), ou a été désactivée (soit en désactivant la directive `enable_dl` ou en activant le `safe mode` dans le `php.ini`) une E_ERROR sera émise et l'exécution du script sera stoppée. Si la fonction `dl` échoue parce que la bibliothèque n'a pu être trouvée, `dl` retournera FALSE et émettra un message d'alerte E_WARNING .

Utilisez la fonction `extension_loaded` pour vérifier qu'une extension est chargée ou non. Cette fonction travaille aussi bien avec les extensions natives qu'avec les extensions dynamiquement chargées (via le `php.ini` ou `dl`).

La fonction `dl` est déprécié depuis PHP 5. Utilisez les directives d'extensions de chargement à la place.

Exemple avec `dl`

```
<?php
// Chargement pour toutes plates-formes
if (!extension_loaded('sqlite')) {
    if (strtoupper(substr(PHP_OS, 0, 3)) == 'WIN') {
        dl('php_sqlite.dll');
    } else {
        dl('sqlite.so');
    }
}

// Mais la constante PHP_SHLIB_SUFFIX est disponible depuis PHP 4.3.0
if (!extension_loaded('sqlite')) {
    $prefix = (PHP_SHLIB_SUFFIX === 'dll') ? 'php_' : '';
    dl($prefix . 'sqlite.' . PHP_SHLIB_SUFFIX);
}
?>
```

Le dossier à partir duquel sont chargées vos extensions dépend de votre plate-forme :

Windows - S'il n'est pas explicitement indiqué dans le fichier `php.ini` , le dossier des extensions est `c:\php4\extensions\` .

Unix - S'il n'est pas explicitement indiqué dans le fichier `php.ini` , le dossier des extensions dépend de

- Si PHP a été compilé avec l'option `--enable-debug` ou non
- Si PHP a été compilé avec le support (expérimental) de ZTS (Zend Thread Safety) ou non
- de la constante interne `ZEND_MODULE_API_NO` (version interne de module d'API Zend, qui est en réalité la date à laquelle une modification importante de l'API a été faite, par exemple 20010901)

En prenant ces paramètres en considération, le dossier des extensions vaut alors `<php-install-directory>/lib/php/extension/<debug-or-not>-<zts-or-not>-ZEND_MODULE_API_NO` , i.e. `/usr/local/php/lib/php/extensions/debug-non-zts-20010901` ou `/usr/local/php/lib/php/extensions/no-debug-zts-20010901` .

Note

`dl` n'est pas supportée sur les serveur web multi-threadés. Utilisez la directive `extensions` dans votre fichier `php.ini` lorsque vous vous trouvez dans un environnement de ce type. Cependant, les versions CGI et CLI **ne sont pas** affectées !

Note

`dl` est sensible à la casse sur les plates-formes Unix.

Note

Cette fonction est désactivée par le `safe-mode`

Voir aussi [directives de chargement d'extensions](#) et [extension_loaded](#) .

8.55.10 `extension_loaded()` : Détermine si une extension est chargée ou non

bool `extension_loaded` (string name)

`extension_loaded` retourne TRUE si l'extension name a été chargée, FALSE sinon.

Exemple avec `extension_loaded`

```
<?php
if (!extension_loaded('gd')) {
    if (!dl('gd.so')) {
        exit;
    }
}
?>
```

Vous pouvez connaître le nom des différentes extensions PHP en utilisant la fonction `phpinfo` ou bien si vous utilisez la version CGI ou CLI de PHP, vous pouvez utiliser l'option de ligne de commande `-m` pour afficher toutes les extensions disponibles :

```
$ php -m
[PHP Modules]
xml
tokenizer
standard
sockets
session
posix
pcre
overload
mysql
mbstring
ctype
```

[Zend Modules]

Note

`extension_loaded` utilise le nom interne de l'extension pour vérifier si une extension est disponible ou pas. La plupart des extensions ont des noms internes écrits en minuscules, mais il peut arriver que certaines aient des noms en majuscules. Soyez donc prévenus que la comparaison effectuée par cette fonction est **sensible à la casse** !

Voir aussi [get_loaded_extensions](#) , [get_extension_funcs](#) , [phpinfo](#) et [dl](#) .

8.55.11 `get_cfg_var()` : Retourne la valeur d'une option de PHP

string `get_cfg_var` (string varname)

`get_cfg_var` retourne la valeur courante de l'option PHP varname ou bien FALSE en cas d'erreur.

`get_cfg_var` ne retourne pas les options qui ont été choisies lors de la compilation de PHP, ni ne lit dans le fichier de configuration d'Apache.

Pour vérifier si le système utilise le fichier de configuration, essayez de lire la valeur de `cfg_file_path`. Si cette valeur est disponible, alors le fichier de configuration est utilisé.

Voir aussi [ini_get](#).

8.55.12 `get_current_user()` : Retourne le nom du possesseur du script courant

string `get_current_user` (void)

`get_current_user` retourne le nom du possesseur du script courant.

Voir aussi [getmyuid](#), [getmygid](#), [getmypid](#), [getmyinode](#) et [getlastmod](#).

8.55.13 `get_defined_constants()` : Retourne la liste des constantes et leurs valeurs

array `get_defined_constants` (*mixed* `categorize`)

`get_defined_constants` retourne les noms et valeurs des constantes déjà définies. Cela inclut les constantes créées par les extensions, et celles créées avec la fonction [define](#).

Exemple avec `get_defined_constants`

```
<?php
print_r(get_defined_constants());
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[E_ERROR] => 1
[E_WARNING] => 2
[E_PARSE] => 4
[E_NOTICE] => 8
[E_CORE_ERROR] => 16
[E_CORE_WARNING] => 32
[E_COMPILE_ERROR] => 64
[E_COMPILE_WARNING] => 128
[E_USER_ERROR] => 256
[E_USER_WARNING] => 512
[E_USER_NOTICE] => 1024
[E_ALL] => 2047
[TRUE] => 1
)
```

Depuis PHP 5, un paramètre additionnel `categorize` peut être passé, permettant à cette fonction de retourner un tableau multi-dimensionnel avec les catégories en tant que clés de la première

dimension et les constantes ainsi que leurs valeurs dans la seconde dimension.

```
<?php
define("MY_CONSTANT", 1);
print_r(get_defined_constants(true));
?>
```

L'exemple ci-dessus va afficher quelque chose de similaire à :

```
Array
(
  [internal] => Array
  (
    [E_ERROR] => 1
    [E_WARNING] => 2
    [E_PARSE] => 4
    [E_NOTICE] => 8
    [E_CORE_ERROR] => 16
    [E_CORE_WARNING] => 32
    [E_COMPILE_ERROR] => 64
    [E_COMPILE_WARNING] => 128
    [E_USER_ERROR] => 256
    [E_USER_WARNING] => 512
    [E_USER_NOTICE] => 1024
    [E_ALL] => 2047
    [TRUE] => 1
  )

  [pcre] => Array
  (
    [PREG_PATTERN_ORDER] => 1
    [PREG_SET_ORDER] => 2
    [PREG_OFFSET_CAPTURE] => 256
    [PREG_SPLIT_NO_EMPTY] => 1
    [PREG_SPLIT_DELIM_CAPTURE] => 2
    [PREG_SPLIT_OFFSET_CAPTURE] => 4
    [PREG_GREP_INVERT] => 1
  )

  [user] => Array
  (
    [MY_CONSTANT] => 1
  )
)
```

Note

La valeur du paramètre `categorize` n'est pas révélatrice, seule sa présence est considérée.

Voir aussi [defined](#) , [get_loaded_extensions](#) , [get_defined_functions](#) et [get_defined_vars](#) .

8.55.14 `get_extension_funcs()` : Liste les fonctions d'une extension

array `get_extension_funcs` (string `module_name`)

`get_extension_funcs` retourne le nom des fonctions définies dans le module `module_name` .

Note

Le paramètre `module_name` doit être en **minuscule** .

Exemple avec `get_extension_funcs`

Ces deux lignes affichent la liste des fonctions disponibles avec les modules `xml` et `gd` .

```
<?php
print_r(get_extension_funcs("xml"));
print_r(get_extension_funcs("gd"));
?>
```

Voir aussi `get_loaded_extensions` .

8.55.15 `get_include_path()` : Lit la valeur de la directive de configuration `include_path`

string `get_include_path` ()

`get_include_path` lit la valeur de la directive de configuration `include_path` .

Exemple avec `get_include_path`

```
<?php
// Fonctionne depuis PHP 4.3.0
echo get_include_path();

// Fonctionne sur toutes les versions
echo ini_get('include_path');
?>
```

Voir aussi `ini_get` , `restore_include_path` , `set_include_path` et `include` .

8.55.16 `get_included_files()` : Retourne un tableau avec les noms des fichiers qui sont inclus dans un script

array `get_included_files` (void)

`get_included_files` retourne un tableau contenant les noms de tous les fichiers qui ont été ajoutés au script avec les fonctions `require_once` , `include_once` , `require` ou `include` .

Le script en cours est considéré comme fichier inclus, il sera donc listé avec les autres fichiers.

Les fichiers inclus ou requis plusieurs fois ne s'affichent qu'une fois dans le tableau retourné.

Note

Les fichiers inclus en utilisant la directive de configuration `auto_prepend_file` ne sont pas listés.

Exemple avec `get_included_files` (abc.php)

```
<?php
include 'test1.php';
include_once 'test2.php';
require 'test3.php';
require_once 'test4.php';

$included_files = get_included_files();

foreach ($included_files as $filename) {
    echo "$filename\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
abc.php
test1.php
test2.php
test3.php
test4.php
```

Note

En PHP 4.0.1pl2, cette fonction supposait que `required_once` utilisait l'extension ".php" : les autres extensions ne fonctionnaient pas. Par ailleurs, dans cette version, le tableau retourné était un tableau associatif et ne reprenait que les fichiers ajoutés avec `include` et `include_once`.

Voir aussi `include`, `include_once`, `require`, `require_once`, et `get_required_files`.

8.55.17 `get_loaded_extensions()` : Retourne la liste de tous les modules compilés et chargés

array `get_loaded_extensions` (void)

`get_loaded_extensions` retourne un tableau contenant les noms de tous les modules compilés et chargés sur l'interpreteur PHP courant.

Par exemple, la ligne ci-dessous

Exemple avec `get_loaded_extensions`

```
<?php
print_r(get_loaded_extensions());
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [0] => xml
    [1] => wddx
    [2] => standard
```

```

[3] => session
[4] => posix
[5] => pgsql
[6] => pcre
[7] => gd
[8] => ftp
[9] => db
[10] => Calendar
[11] => bcmath
)

```

Voir aussi [get_extension_funcs](#) , [extension_loaded](#) , [dl](#) et [phpinfo](#) .

8.55.18 [get_magic_quotes_gpc\(\)](#) : Retourne la configuration actuelle de l'option [magic_quotes_gpc](#)

```
int get\_magic\_quotes\_gpc ( void )
```

[get_magic_quotes_gpc](#) retourne la configuration actuelle de l'option [magic_quotes_gpc](#) (0 pour l'option désactivée, 1 pour l'option activée).

Note

Si la directive [magic_quotes_sybase](#) est activée, elle remplacera complètement [magic_quotes_gpc](#) . Ce qui fait que même si [get_magic_quotes](#) retourne TRUE les guillemets doubles, les anti-slashes ou les caractères NULL ne seront pas protégés. Seul les guillemets simples le seront. Dans ce cas, ils ressembleront à " .

Gardez en tête que la configuration de [magic_quotes_gpc](#) ne fonctionnera pas durant l'exécution du script.

Exemple avec [get_magic_quotes_gpc](#)

```

<?php
echo get\_magic\_quotes\_gpc();           // 1
echo $_POST['lastname'];                 // O\'reilly
echo addslashes($_POST['lastname']);     // O\\\'reilly

if (!get\_magic\_quotes\_gpc()) {
    $lastname = addslashes($_POST['lastname']);
} else {
    $lastname = $_POST['lastname'];
}

echo $lastname; // O\'reilly
$sql = "INSERT INTO lastnames (lastname) VALUES ('$lastname)";
?>

```

Pour plus d'informations sur [magic_quotes](#), voir la section sur les [guillemets magiques](#) .

Voir aussi [addslashes](#) , [stripslashes](#) , [get_magic_quotes_runtime](#) et [ini_get](#) .

8.55.19 [get_magic_quotes_runtime\(\)](#) : Retourne la configuration actuelle de l'option [magic_quotes_runtime](#)

```
int get\_magic\_quotes\_runtime ( void )
```

[get_magic_quotes_runtime](#) retourne la configuration actuelle de l'option [magic_quotes_runtime](#) . (0 pour option désactivée, 1 pour option activée).

Voir aussi [get_magic_quotes_gpc](#) et [set_magic_quotes_runtime](#) .

8.55.20 [get_required_files\(\)](#) : Alias de [get_included_files](#)

Cette fonction est un alias de : [get_included_files](#) .

8.55.21 [getenv\(\)](#) : Retourne la valeur d'une variable d'environnement

string **getenv** (string varname)

[getenv](#) retourne la valeur de la variable d'environnement varname , ou FALSE en cas d'erreur.

Exemple avec [getenv](#)

```
<?php
// Exemple d'utilisation de getenv()
$ip = getenv('REMOTE_ADDR');

// Ou utilisez simplement une Superglobale ($_SERVER ou $_ENV)
$ip = $_SERVER['REMOTE_ADDR'];
?>
```

Vous pouvez voir une liste complète des variables d'environnement en utilisant la fonction [phpinfo](#) . Vous pouvez trouver la signification de chacune d'entre elles en consultant le site concernant [CGI specification](#) (en anglais), et particulièrement la page concernant les [variables d'environnement](#) .

Voir aussi [putenv](#) , [apache_getenv](#) et les [Superglobales](#) .

8.55.22 [getlastmod\(\)](#) : Retourne la date de dernière modification de la page

int **getlastmod** (void)

[getlastmod](#) retourne la date de dernière modification de la page. La valeur retournée est un timestamp UNIX, utilisable comme paramètre avec la fonction [date](#) . [getlastmod](#) retourne FALSE en cas d'erreur.

Exemple avec [getlastmod](#)

```
<?php
// affiche par exemple 'Dernière modification: April 20 2004 20:43:59.'
echo 'Dernière modification : ' . date( "F d Y H:i:s.", getlastmod());
?>
```

Note

Si vous voulez récupérer la date de la dernière modification d'un fichier différent, utilisez la fonction [filemtime](#) .

Voir aussi [date](#) , [getmyuid](#) , [getmygid](#) , [get_current_user](#) , [getmyinode](#) , [getmypid](#) et [filemtime](#) .

8.55.23 `getmygid()` : Retourne le GID du propriétaire du script

int `getmygid` (void)

`getmygid` retourne l'identifiant du groupe du propriétaire du script (GID), ou FALSE en cas d'erreur.

Voir aussi `getmyuid` , `getmypid` , `get_current_user` , `getmyinode` et `getlastmod` .

8.55.24 `getmyinode()` : Retourne l'inode du script

int `getmyinode` (void)

`getmyinode` retourne l'inode du script, ou FALSE en cas d'erreur.

Voir aussi `getmygid` , `getmyuid` , `get_current_user` , `getmypid` et `getlastmod` .

Note

Cette fonction n'est pas implémentée sous Windows.
--

8.55.25 `getmypid()` : Retourne le numéro de processus courant de PHP

int `getmypid` (void)

`getmypid` retourne le numéro de processus actuel ou FALSE en cas d'erreur.

Attention

Les identifiants de processus ne sont pas uniques, et forment une source d'entropie faible. Nous recommandons de ne pas utiliser les pid pour assurer la sécurité d'un système.

Voir aussi `getmygid` , `getmyuid` , `get_current_user` , `getmyinode` et `getlastmod` .

8.55.26 `getmyuid()` : Retourne l'UID du propriétaire du script actuel

int `getmyuid` (void)

`getmyuid` retourne l'identifiant du propriétaire du script actuel (UID) ou FALSE en cas d'erreur.

Voir aussi `getmygid` , `getmypid` , `get_current_user` , `getmyinode` et `getlastmod` .

8.55.27 `getopt()` : Lit des options passées dans la ligne de commande

array `getopt` (string options , array longopts)

`getopt` retourne un tableau associatif avec la liste des paires option/valeur, lues dans la ligne de commande, et basée sur le format d'option options ou bien FALSE en cas d'erreur.

Sur les plates-formes qui ont la fonction C `getopt_long` , les options longues peuvent être spécifiées avec le paramètre `longopts` (disponible depuis PHP 4.3.0).

Exemple avec <code>getopt</code>

```
<?php
$options = getopt("f:hp:");
// Analyse la ligne de commande (i.e. $GLOBALS['argv'])
?>
```

Le paramètre options peut contenir les éléments suivants : caractères ou caractères suivis d'un signe deux-points pour indiquer un élément optionnel. Par exemple, une option appelée x correspond à une option -x , et une chaîne x: correspond à une option de type -x argument . Cela importe peu si un argument a des espaces auparavant.

`getopt` retourne un tableau avec les options et leurs valeurs. Si une option n'a pas d'argument, sa valeur sera FALSE .

Note

Cette fonction n'est pas implémentée sous Windows.

8.55.28 `getrusage()` : Retourne le niveau d'utilisation des ressources

array `getrusage` (*int who*)

`getrusage` est une interface à la fonction system `getrusage(2)`. Elle retourne un tableau associatif contenant les informations renvoyées par cet appel système. Si `who` est égal à 1, `getrusage` sera appelé avec le paramètre `RUSAGE_CHILDREN`.

Toutes les valeurs du tableau sont accessibles en utilisant leur nom dans le tableau.

Exemple avec `getrusage`

```
<?php
$dat = getrusage();
echo $dat["ru_nswap"];           // Taille de la mémoire swap
echo $dat["ru_majflt"];         // Nombre de page mémoires utilisées
echo $dat["ru_utime.tv_sec"];   // Temps utilisateur (en secondes)
echo $dat["ru_utime.tv_usec"]; // Temps utilisateur (en microsecondes)
?>
```

Consultez la page de manuel `getrusage(2)` de votre système pour plus de détails.

Note

Cette fonction n'est pas implémentée sous Windows.

8.55.29 `ini_alter()` : Alias de `ini_set`

Cette fonction est un alias de : `ini_set` .

8.55.30 `ini_get_all()` : Lit toutes les valeurs de configuration

array `ini_get_all` (*string extension*)

`ini_get_all` retourne toutes les valeurs de configuration sous la forme d'un tableau associatif. Si le paramètre optionnel `extension` est fourni, `ini_get_all` retourne uniquement les configurations concernant cette extension.

Le tableau retourné utilise les noms des directives comme index, et les valeurs associées sont `global_value` (configuré dans `php.ini`), `local_value` (configuré via `ini_set` ou `.htaccess`), et `access` (le niveau d'accès). Voyez le manuel sur les [changements de configuration](#) pour plus d'informations

sur les niveaux d'accès.

Note

Il est possible pour une directive d'avoir plusieurs niveaux d'accès, ce qui est indiqué dans le champ de bits access .

Exemple avec <code>ini_get_all</code>

```
<?php
$inis = ini_get_all();

print_r($inis);
?>
```

L'affichage de ce script peut être :

```
Array
(
[allow_call_time_pass_reference] => Array
(
[global_value] => 1
[local_value] => 1
[access] => 6
)
[allow_url_fopen] => Array
(
[global_value] => 1
[local_value] => 1
[access] => 7
)
...
)
```

Voir aussi [get_loaded_extensions](#) , [phpinfo](#) , [ini_restore](#) , [ini_get](#) et [ini_set](#)

8.55.31 `ini_get()` : Lit la valeur d'une option de configuration

string `ini_get` (string varname)

`ini_get` retourne la valeur de l'option de configuration varname en cas de succès, et FALSE sinon.

Note

Lecture de valeurs booléennes

Une directive de configuration ayant la valeur de off sera retourné sous la forme d'une chaîne vide ou "0" alors que la valeur on retournera " -1 ".
--

Note

Lors de la lecture des tailles de mémoire

Plusieurs directives traitant de taille mémoire, comme <code>upload_max_filesize</code> , sont stockées dans le fichier <code>php.ini</code> avec une notation courte. <code>ini_get</code> retourne la chaîne exacte stockée dans le fichier

php.ini et **NON PAS** son équivalent entier. Appliquer des opérations arithmétiques classiques sur ces valeurs ne conduira à rien de bon. L'exemple ci-dessous montre une façon de convertir la notation sténographique en octets, de la même façon dont le fait le source PHP.

Les directives liées aux tailles mémoires sont lues littéralement

```
<?php
/*
Notre fichier php.ini contient les directives suivantes :

display_errors = On
register_globals = Off
post_max_size = 8M
*/

echo 'display_errors = ' . ini_get('display_errors') . "\n";
echo 'register_globals = ' . ini_get('register_globals') . "\n";
echo 'post_max_size = ' . ini_get('post_max_size') . "\n";
echo 'post_max_size+1 = ' . (ini_get('post_max_size')+1) . "\n";
echo 'post_max_size en octets = ' . return_bytes(ini_get('post_max_size'));

function return_bytes($val) {
    $val = trim($val);
    $last = strtolower($val{strlen($val)-1});
    switch($last) {
        // Le modifieur 'G' est disponible depuis PHP 5.1.0
        case 'g':
            $val *= 1024;
        case 'm':
            $val *= 1024;
        case 'k':
            $val *= 1024;
    }
    return $val;
}
?>
```

L'exemple ci-dessus va afficher :

```
display_errors = 1
register_globals = 0
post_max_size = 8M
post_max_size+1 = 9
post_max_size in bytes = 8388608
```

Voir aussi [get_cfg_var](#) , [ini_get_all](#) , [ini_restore](#) et [ini_set](#) .

8.55.32 ini_restore() : Restaure la valeur de l'option de configuration

```
void ini_restore ( string varname )
```

[ini_restore](#) restaure la valeur originale de l'option de configuration varname .

Voir aussi [ini_get](#) , [ini_get_all](#) et [ini_set](#) .

8.55.33 ini_set() : Modifie la valeur d'une option de configuration

string **ini_set** (string varname , string newvalue)

ini_set change la valeur de l'option de configuration varname et lui donne celle de newvalue . ini_set retourne FALSE en cas d'échec, et la valeur précédente en cas de succès. La valeur de l'option de configuration sera modifiée durant toute l'exécution du script et pour ce script spécifiquement. Elle reprendra sa valeur par défaut dès la fin du script.

Les options disponibles ne peuvent pas toutes être modifiées avec ini_set . La liste de toutes les options disponibles se trouve dans l' annexe .

Voir aussi get_cfg_var , ini_get , ini_get_all , ini_restore et Comment modifier la configuration

8.55.34 main() : Fausse documentation pour main

Il n'y a pas de fonction appelée main , hormis dans les sources PHP. En PHP 4.3.0, un nouveau type de gestion d'erreur a été introduit dans les sources de PHP : php_error_docref. Une des fonctionnalités était de fournir un lien vers la page de manuel de PHP lorsque les directives html_errors (par défaut) et docref_root (par défaut jusqu'en PHP 4.3.2) sont activées.

Certains messages d'erreur faisaient référence à une page de manuel pour la fonction main alors que cette page n'existe pas. Ajoutez une note utilisateur, en anglais, où vous mentionnez la fonction PHP vous ayant conduite sur cette page. Cette erreur sera corrigée et documentée.

Nom de la fonction	Ne pointe plus sur cette page depuis la version
<u>include</u>	5.1.0
<u>include_once</u>	5.1.0
<u>require</u>	5.1.0
<u>require_once</u>	5.1.0

Voir aussi html_errors et display_errors .

8.55.35 memory_get_usage() : Indique la quantité de mémoire utilisée par PHP

int **memory_get_usage** ()

memory_get_usage retourne la quantité de mémoire allouée à PHP à cet instant.

memory_get_usage n'est disponible que si PHP est compilé avec l'option --enable-memory-limit .

Exemple avec memory_get_usage

```

<?php
// Ceci n'est qu'un exemple. Les chiffres ci-dessous
// différeront suivant les systèmes et les configurations

echo memory_get_usage() . "\n"; // 36640

$a = str_repeat("Salut", 4242);

echo memory_get_usage() . "\n"; // 57960

```

```
unset($a);

echo memory_get_usage() . "\n"; // 36744

?>
```

Voir aussi [memory-limit](#) .

8.55.36 `php_ini_scanned_files()` : Retourne la liste des fichiers .ini analysés dans les dossiers de configuration supplémentaires

string `php_ini_scanned_files` (void)

`php_ini_scanned_files` retourne une liste de noms de fichiers de configuration analysés, en plus de `php.ini` . Cette liste est au format CSV. Ces fichiers sont situés dans un dossier défini par l'option `--with-config-file-scan-dir` , définie au moment de la compilation.

`php_ini_scanned_files` retourne une chaîne de caractères, où les noms de fichiers sont séparés par des virgules. Si `--with-config-files-scan-dir` n'était pas configuré, `FALSE` est retourné. Si cette option était configurée, mais que le dossier était vide, une chaîne vide est retournée. Si un fichier n'est pas lisible, le nom du fichier sera inclus dans la liste, mais une erreur sera générée. Cette erreur sera visible au moment de la compilation, et lorsque vous appellerez la fonction `php_ini_scanned_files` .

Les fichiers retournés incluent le chemin, comme déclaré dans la directive `--with-config-file-scan-dir` . De plus, chaque virgule est suivie d'une nouvelle ligne.

Un exemple de liste retournée par `php_ini_scanned_files`

```
<?php

if ($filelist = php_ini_scanned_files()) {
    if (strlen($filelist) > 0) {
        $files = explode(',', $filelist);
        echo "<ul>";
        foreach ($files as $file) {
            echo "<li>" . trim($file) . "</li>\n";
        }
        echo "</ul>";
    }
}

?>
```

Voir aussi [ini_set](#) et [phpinfo](#) .

8.55.37 `php_logo_guid()` : Retourne l'identifiant du logo PHP

string `php_logo_guid` (void)

Cette fonction retourne l'identifiant qui peut être utilisé pour afficher le logo PHP en utilisant une image incorporée dans les sources. Le logo est affiché uniquement si `expose_php` vaut `On` .

Exemple avec `php_logo_guid`

```
<?php

echo '';
```

```
?>
```

Voir aussi [phpinfo](#) , [phpversion](#) , [phpcredits](#) et [zend_logo_guid](#) .

8.55.38 `php_sapi_name()` : Retourne le type d'interface utilisée entre le serveur web et PHP

string `php_sapi_name` (void)

`php_sapi_name` retourne une chaîne en minuscule qui décrit le type d'interface utilisée entre le serveur web et PHP (Server API, SAPI). En CGI PHP, cette chaîne est "cgi", en mod_php pour Apache, cette chaîne est "apache", etc.

Exemple avec `php_sapi_name`

```
<?php
$inter_type = php_sapi_name();
if (substr($sapi_type, 0, 3) == 'cgi') {
    echo "Vous utilisez CGI PHP\n";
} else {
    echo "Vous n'utilisez pas CGI PHP\n";
}
?>
```

8.55.39 `php_uname()` : Retourne les informations sur le système d'exploitation

string `php_uname` (*string mode*)

`php_uname` retourne une description sur le système d'exploitation sur lequel tourne PHP. Si vous voulez juste savoir le nom du système d'exploitation, utilisez plutôt la constante `PHP_OS` mais gardez à l'esprit que cette constante contient le nom du système sur lequel PHP a été **compilé** .

Sous l'Unix, la fonction tente d'afficher les informations du système d'exploitation sur lequel PHP a été compilé si elle n'arrive pas à déterminer le système d'exploitation courant.

mode est un seul caractère qui définit quelles seront les informations à retourner :

- 'a' : Par défaut. Contient tous les modes de la séquence "s n r v m" .
- 's' : Nom du système d'exploitation. eg. FreeBSD .
- 'n' : Nom de l'hôte. eg. localhost.example.com .
- 'r' : Nom de la version. eg. 5.1.2-RELEASE .
- 'v' : Information sur la version. Varie énormément suivant le système d'exploitation.
- 'm' : Type de la machine. eg. i386 .

Exemples avec `php_uname`

```
<?php
echo php_uname();
echo PHP_OS;

/* Affichages possibles :
Linux localhost 2.4.21-0.13mdk #1 Fri Mar 14 15:08:06 EST 2003 i686
Linux
```

```
FreeBSD localhost 3.2-RELEASE #15: Mon Dec 17 08:46:02 GMT 2001
FreeBSD

Windows NT XN1 5.1 build 2600
WINNT
*/

if (strtoupper(substr(PHP_OS, 0, 3)) === 'WIN') {
    echo 'Le serveur tourne sous Windows !';
} else {
    echo 'Le serveur ne tourne pas sous Windows !';
}

?>
```

Il existe aussi des constantes PHP pré-définies liées qui peuvent s'avérer utiles, par exemple :

Exemples avec quelques constantes liées au système

```
<?php
// *nix
echo DIRECTORY_SEPARATOR; // affiche : "/"
echo PATH_SEPARATOR;      // affiche : ":"
echo PHP_SHLIB_SUFFIX;    // affiche : "so"

// Win*
echo DIRECTORY_SEPARATOR; // affiche : "\"
echo PATH_SEPARATOR;      // affiche : ";"
echo PHP_SHLIB_SUFFIX;    // affiche : "dll"
?>
```

Voir aussi [phpversion](#) , [php_sapi_name](#) et [phpinfo](#) .

8.55.40 `phpcredits()` : Affiche les crédits de PHP

`bool phpcredits (int flag)`

`phpcredits` affiche la liste des développeurs PHP, des modules, etc. Elle génère le code HTML approprié pour insérer les informations dans une page. Le paramètre `flag` indique les informations qui doivent être affichées. Par exemple, pour afficher les crédits généraux, vous pouvez utiliser le code suivant :

Exemple avec `phpcredits`

```
<?php
phpcredits(CREDITS_GENERAL);
?>
```

Et pour afficher la liste des développeurs et du groupe de documentation dans une page séparée, vous utiliserez

Exemple avec `phpcredits` (2)

```
<?php
phpcredits(CREDITS_GROUP + CREDITS_DOCS + CREDITS_FULLPAGE);
?>
```

Si vous vous sentez l'envie de placer tous les crédits dans votre page, vous pouvez utiliser ceci :

Personnalisation de la page de crédits PHP

```
<html>
```

```

<head>
  <title>Ma page de crédits</title>
</head>
<body>
<?php
  // Un peu de votre code
  phpcredits(CREDITS_ALL - CREDITS_FULLPAGE);
  // Encore un peu de votre code
?>
</body>
</html>

```

Nom	Description
CREDITS_ALL	Tous les crédits, équivalent à : CREDITS_DOCS + CREDITS_GENERAL + CREDITS_GROUP + CREDITS_MODULES + CREDITS_FULLPAGE . La fonction génère alors une page HTML complète.
CREDITS_DOCS	Les crédits du groupe de documentation
CREDITS_FULLPAGE	En général, ce paramètre est utilisé avec d'autres constantes. Il indique que la page ainsi générée doit être une page HTML complète, avec toutes les balises nécessaires.
CREDITS_GENERAL	Crédits Généraux : conception et design du langage, auteurs de PHP 4.0, module SAPI.
CREDITS_GROUP	Une liste des développeurs principaux
CREDITS_MODULES	Une liste des extensions de PHP, et leurs auteurs
CREDITS_SAPI	Une liste des modules Server API pour PHP et leurs auteurs.

Voir aussi [phpinfo](#) , [phpversion](#) et [php_logo_guid](#) .

8.55.41 [phpinfo\(\)](#) : Affiche de nombreuses informations sur PHP

bool **phpinfo** (*int what*)

[phpinfo](#) affiche de nombreuses informations sur PHP, concernant sa configuration courante : options de compilation, extensions, version, informations sur le serveur, et l'environnement (lorsqu'il est compilé comme module), environnement PHP, informations sur le système, chemins, valeurs générales et locales de configuration, en-têtes HTTP et la licence PHP.

Comme tous les systèmes sont configurés différemment, [phpinfo](#) sert généralement à vérifier la [configuration](#) ainsi que les [variables pré-définies](#) , pour une plate-forme donnée. De plus, [phpinfo](#) est un bon outil de débogage, car il affiche le contenu de toutes les variables EGPCS (Environnement, GET, POST, Cookie, Serveur).

L'affichage peut être personnalisé en utilisant une ou plusieurs des constantes suivantes. Elles sont combinables avec l'opérateur **OR** , et doivent être passées dans le paramètre *what* . Vous pouvez aussi additionner ces constantes.

Nom de la constante	Valeur	Description
INFO_GENERAL	1	La ligne de configuration, le chemin du php.ini , la date de compilation, le serveur web, le système, etc.
INFO_CREDITS	2	Les crédits de PHP. Voir aussi phpcredits .
INFO_CONFIGURATION	4	Valeurs courantes locales et générales des directives PHP. Voyez aussi la fonction ini_get .
INFO_MODULES	8	

		Modules chargés et leur configuration spécifique. Voir aussi la fonction get_loaded_extensions .
INFO_ENVIRONMENT	16	Informations sur les variables d'environnement, qui sont disponibles dans la variable <code>\$_ENV</code> .
INFO_VARIABLES	32	Affiche toutes les variables pré-définies , issues de l'environnement, la méthode GET, la méthode POST, les cookies et le serveur.
INFO_LICENSE	64	La licence PHP. Voir aussi la FAQ de la licence .
INFO_ALL	-1	Affiche toutes les informations suscitées. C'est la valeur par défaut.

Exemple avec [phpinfo](#)

```
<?php
// Affiche toutes les informations, comme le ferait INFO_ALL
phpinfo();

// Affiche uniquement le module d'information.
// phpinfo(8) fournirait les mêmes informations.
phpinfo(INFO_MODULES);

?>
```

Note

Une partie des informations affichées sont désactivées si la directive [expose_php](#) est configurée avec la valeur off . Cela inclus les logos PHP et Zend, ainsi que les crédits.

Note

[phpinfo](#) affiche du texte au lieu de HTML lorsque vous utilisez la version CLI.

Voir aussi [phpversion](#) , [phpcredits](#) , [php_logo_guid](#) , [ini_get](#) , [ini_set](#) , [get_loaded_extensions](#) et la section sur les [variables pré-définies](#) .

8.55.42 [phpversion\(\)](#) : Retourne le numéro de la version courante de PHP

string [phpversion](#) (*string extension*)

[phpversion](#) retourne le numéro de la version courante de PHP. Si le paramètre optionnel *extension* est spécifié, la fonction [phpversion](#) retournera la version de cette extension ou FALSE s'il n'y a aucune information sur la version d'associée ou si cette extensions n'est pas disponible.

Note

Cette information est aussi disponible via la constante prédéfinie `PHP_VERSION` .

Exemple avec [phpversion](#)

```
<?php
// affiche le numéro de version courante du PHP.
echo "Version PHP courante : " . phpversion();

// affiche e.g. '2.0' ou rien du tout si cette extension n'est pas active
echo phpversion('tidy');

?>
```

Voir aussi [version_compare](#) , [phpinfo](#) , [phpcredits](#) , [php_logo_guid](#) et [zend_version](#) .

8.55.43 putenv() : Fixe la valeur d'une variable d'environnement

bool **putenv** (string setting)

putenv fixe la valeur d'une variable d'environnement. Cette valeur n'existera que durant la vie du script courant, et l'environnement initial sera restauré lorsque le script sera terminé.

Modifier la valeur de certaines variables système peut être un trou de sécurité considérable. La directive de configuration `safe_mode_allowed_env_vars` contient une liste de préfixes, séparés par des virgules. Lorsque le Safe Mode est actif, l'utilisateur ne peut que modifier les variables dont le nom commence par les préfixes fournis par cette directive. Par défaut, les utilisateurs ne peuvent modifier que les variables qui commencent par `PHP_` (i.e. `PHP_FOO=BAR`). Note : Si cette directive est vide, PHP autorisera la modification de TOUTES les variables d'environnement !

La directive de configuration `safe_mode_protected_env_vars` contient une liste de variables d'environnement, séparées par des virgules. Les utilisateurs ne pourront pas modifier ces variables avec la fonction **putenv**. Ces variables seront protégées même si `safe_mode_allowed_env_vars` permet leur modification.

Attention

Ces directives n'ont d'effets que lorsque le **safe mode** est activé !

Modification d'une variable d'environnement

```
<?php
putenv("UNIQID=$uniqid");
?>
```

Voir aussi [getenv](#) .

8.55.44 restore_include_path() : Restaure la valeur de la directive de configuration include_path

void **restore_include_path** ()

restore_include_path restaure la valeur de la directive de configuration `include_path` à sa valeur originale de début de script, telle qu'inscrite dans le `php.ini` .

Exemple `restore_include_path`

```
<?php
echo get_include_path(); // ./usr/local/lib/php
set_include_path('/inc');
echo get_include_path(); // /inc

// Fonctionne depuis PHP 4.3.0
restore_include_path();

// Fonctionne sur toutes les versions
ini_restore('include_path');

echo get_include_path(); // ./usr/local/lib/php
?>
```

Voir aussi [ini_restore](#) , [set_include_path](#) , [get_include_path](#) et [include](#) .

8.55.45 [set_include_path\(\)](#) : Modifie la valeur de la directive de configuration [include_path](#)

string [set_include_path](#) (string new_include_path)

[set_include_path](#) modifie la valeur de la directive de configuration [include_path](#) , pour la durée du script en cours. Elle retourne l'ancienne valeur de [include_path](#) en cas de succès, et FALSE sinon.

Exemple avec [set_include_path](#)

```
<?php
// Fonctionne depuis PHP 4.3.0
set_include_path('/inc');

// Fonctionne sur toutes les versions
ini_set('include_path', '/inc');
?>
```

Ajout dans le chemin d'inclusion

En utilisant la constante `PATH_SEPARATOR` , il est possible d'étendre le chemin d'inclusion au vu du système.

Dans cet exemple, nous ajoutons `/usr/lib/pear` à la fin de l'actuel `include_path` .

```
<?php
$path = '/usr/lib/pear';
set_include_path(get_include_path() . PATH_SEPARATOR . $path);
?>
```

Voir aussi [ini_set](#) , [get_include_path](#) , [restore_include_path](#) et [include](#) .

8.55.46 [set_magic_quotes_runtime\(\)](#) : Active/désactive l'option [magic_quotes_runtime](#)

bool [set_magic_quotes_runtime](#) (int new_setting)

[set_magic_quotes_runtime](#) active/désactive l'option [magic_quotes_runtime](#) . (0 l'option est désactivée, 1 l'option est activée).

Voir aussi [get_magic_quotes_gpc](#) et [get_magic_quotes_runtime](#) .

8.55.47 [set_time_limit\(\)](#) : Fixe le temps maximum d'exécution d'un script

void [set_time_limit](#) (int seconds)

[set_time_limit](#) fixe le délai d'expiration d'un script, en secondes. Si cette limite est atteinte, le script s'interrompt, et renvoie une erreur fatale. La valeur par défaut est 30 secondes ou, si c'est le cas, la valeur de la directive `max_execution_time` définie dans le `php.ini` . Si la valeur est zéro, il n'y a alors aucune limite imposée.

8.55.44 [restore_include_path\(\)](#) : Restaure la valeur de la directive de configuration `include_path` 941

Lorsqu'elle est appelée, `set_time_limit` remet le compteur à zéro. En d'autres termes, si la limite par défaut est à 30 secondes, et qu'après 25 secondes d'exécution du script l'appel `set_time_limit(20)` est fait, alors le script tournera pendant un total de 45 secondes avant de finir.

Attention

Notez que `set_time_limit` n'a pas d'effet lorsque PHP fonctionne en mode `safe mode`. Il n'y a pas d'autre solution que de changer de mode, ou de modifier la durée maximale d'exécution dans le `php.ini`.

Note

La fonction `set_time_limit` et la directive de configuration `max_execution_time` n'affectent que le temps d'exécution du script lui-même. Tout temps passé en dehors du script, comme un appel système utilisant `system`, des opérations sur les flux, les requêtes sur base de données, etc. n'est pas pris en compte lors du calcul de la durée maximale d'exécution du script.

Voir aussi les directives `max_execution_time` et `max_input_time`.

8.55.48 `version_compare()` : Compare deux chaînes de versions au format des versions PHP

mixed `version_compare` (string `version1` , string `version2` , string `operator`)

`version_compare` compare les deux versions de PHP standardisées. Cette fonction est pratique pour les programmes qui doivent vérifier la version de PHP qui les fait tourner.

`version_compare` retourne -1 si `version1` est inférieure à `version2`, 0 si elles sont égales, et +1 dans le reste des cas.

`version_compare` remplace dans un premier temps `_`, `-` et `+` par un point (`.`) dans les chaînes de version et insère aussi des points avant et après tout caractère non-numérique pour que, par exemple, `'4.3.5RC1'` devienne `'4.3.5.RC.1'`. Ensuite, elle découpe les résultats comme si vous utilisiez `explode('.', $ver)`. Puis, elle compare les morceaux en allant de gauche à droite. Si une part contient des caractères alphabétiques, ils sont gérés dans l'ordre suivant : `dev < alpha = a < beta = b < RC < pl`. De cette façon, vous pouvez comparer non seulement des versions de différents niveaux, comme `'4.1'` et `'4.1.2'`, mais aussi des versions de développement à la mode de PHP, à n'importe quel stade.

Si vous spécifiez le troisième argument optionnel `operator`, vous pouvez tester une relation particulière. Les opérateurs possibles sont : `<`, `lt`, `<=`, `le`, `>`, `gt`, `>=`, `ge`, `==`, `=`, `eq`, `!=`, `<>`, `ne`. En utilisant cet argument, la fonction retournera TRUE si la relation est vérifiée et FALSE sinon.

Note

La constante `PHP_VERSION` contient la version courante de PHP.

Exemple avec `version_compare`

```
<?php
echo version_compare("4.0.4", "4.0.6"); // -1
echo version_compare("4.0.4", "4.0.6", "<"); // 1
echo version_compare("4.0.6", "4.0.6", "eq"); // 1
?>
```

8.55.49 `zend_logo_guid()` : Retourne le logo de Zend

string `zend_logo_guid` (void)

`zend_logo_guid` retourne l'identifiant pouvant être utilisé pour afficher le logo Zend en utilisant l'image intégrée.

Exemple avec `zend_logo_guid`

```
<?php
echo '';
?>
```

Voir aussi [php_logo_guid](#) .

8.55.50 `zend_version()` : Lit la version courante du moteur Zend

string `zend_version` (void)

`zend_version` retourne une chaîne contenant le numéro de version du moteur d'analyse Zend actuellement en cours d'utilisation.

Exemple avec `zend_version`

```
<?php
// affiche e.g. 'Version du moteur Zend: 1.3.0'
// ou bien quelque chose d'approchant si votre version de PHP date un peu
echo "Version du moteur Zend: "; zend_version();
?>
```

Voir aussi [phpinfo](#) , [phpcredits](#) , [php_logo_guid](#) et [phpversion](#) .

8.56 Ingres II

8.56.1 Introduction

Ces fonctions permettent l'accès à un serveur de base de données Ingres II.

Note

Si vous avez déjà utilisé des extensions PHP permettant l'accès à d'autres serveurs de bases de données, notez qu'Ingres n'accepte pas de requêtes et/ou de transactions concurrentes sur la même connexion, et donc vous ne trouverez aucun identifiant de résultat ou de transaction dans cette extension. Le résultat d'une requête doit être traité avant d'envoyer une autre requête, et une transaction doit être validée ("commit") ou annulée ("roll back") avant de pouvoir en ouvrir une nouvelle (l'ouverture de transaction est faite automatiquement à l'envoi de la première requête).

8.56.2 Pré-requis

Pour compiler PHP avec le support Ingres, vous aurez besoin de la bibliothèque ouverte et des fichiers d'en-têtes fournis avec Ingres II.

8.56.3 Installation

Cette extension PECL n'est pas intégrée à PHP. Des informations comme les notes sur les nouvelles versions, les téléchargements, les sources des fichiers, les informations concernant les mainteneurs ainsi qu'un CHANGELOG, peuvent être trouvées ici :

<http://pecl.php.net/package/ingres> .

Pour pouvoir utiliser ces fonctions, vous devez compiler PHP avec le support Ingres, en utilisant l'option `--with-ingres[=DIR]` où DIR est le répertoire d'installation du serveur Ingres (par défaut, `/II/ingres`). Ceci nécessite les fichiers de bibliothèque de l'en-tête d'Open API qui sont inclus dans Ingres II. Si la variable d'environnement `II_SYSTEM` n'est pas correctement initialisée, vous devrez utiliser `--with-ingres=DIR` , pour spécifier le répertoire où a été installé Ingres.

Lorsque cette extension est utilisée avec Apache, si Apache ne démarre pas et émet l'erreur " PHP Fatal error: Unable to start ingres_ii module in Unknown on line 0 ", assurez-vous que la variable d'environnement `II_SYSTEM` est correctement initialisée. Il suffit souvent d'ajouter " `export II_SYSTEM="/home/ingres/II` " dans le script qui démarre Apache, juste avant le lancement de `httpd` .

8.56.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable	Historique
<code>ingres.allow_persistent</code>	"1"	<code>PHP_INI_SYSTEM</code>	Disponible depuis PHP 4.0.2.
<code>ingres.max_persistent</code>	"-1"	<code>PHP_INI_SYSTEM</code>	
<code>ingres.max_links</code>	"-1"	<code>PHP_INI_SYSTEM</code>	Disponible depuis PHP 4.0.2.
<code>ingres.default_database</code>	NULL	<code>PHP_INI_ALL</code>	Disponible depuis PHP 4.0.2.

ingres.default_user	NULL	PHP_INI_ALL	Disponible depuis PHP 4.0.2.
ingres.default_password	NULL	PHP_INI_ALL	Disponible depuis PHP 4.0.2.
ingres.report_db_warnings	"1"	PHP_INI_ALL	Disponible depuis la version 1.1.0 de l'extension PECL.
ingres.cursor_mode	"0"	PHP_INI_ALL	Disponible depuis la version 1.1.0 de l'extension PECL.
ingres.blob_segment_length	"4096"	PHP_INI_ALL	Disponible depuis la version 1.2.0 de l'extension PECL.

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [Directives du php.ini](#) .

8.56.5 Types de ressources

Cette extension ne définit aucune ressource.

8.56.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

INGRES_ASSOC ([entier](#))

Des colonnes sont retournées dans un tableau ayant comme index le nom des champs.

INGRES_NUM ([entier](#))

Des colonnes sont retournées dans un tableau ayant un index numérique des champs. L'index commence avec 1, le premier champ dans le résultat.

INGRES_BOTH ([entier](#))

Des colonnes sont retournées dans un tableau ayant un index numérique et un index avec le nom des champs.

INGRES_EXT_VERSION ([chaîne de caractères](#))

Spécifie la version de l'extension Ingres. Disponible depuis la version 1.2.0 de l'extension PECL.

INGRES_API_VERSION ([entier](#))

Spécifie la version de Ingres OpenAPI de l'extension qui a été bâtie. Disponible depuis la version 1.2.0 de l'extension PECL.

INGRES_CURSOR_READONLY ([entier](#))

Spécifie que les curseurs Ingres doivent être ouverts en mode lecture seule ('readonly'). Disponible depuis la version 1.2.0 de l'extension PECL.

INGRES_CURSOR_UPDATE ([entier](#))

Spécifie que les curseurs Ingres doivent être ouverts en mode pour mise à jour ('for update'). Disponible depuis la version 1.2.0 de l'extension PECL.

INGRES_DATE_MULTINATIONAL ([entier](#))

Équivalent à la configuration II_DATE_FORMAT de MULTINATIONAL. Disponible depuis la version 1.2.0 de l'extension PECL.

INGRES_DATE_MULTINATIONAL4 ([entier](#))

Équivalent à la configuration II_DATE_FORMAT de MULTINATIONAL4. Disponible depuis la version 1.2.0 de l'extension PECL.

INGRES_DATE_FINNISH ([entier](#))

Équivalent à la configuration II_DATE_FORMAT de FINNISH. Disponible depuis la version 1.2.0 de l'extension PECL.

INGRES_DATE_ISO ([entier](#))

Équivalent à la configuration `II_DATE_FORMAT` de ISO. Disponible depuis la version 1.2.0 de l'extension PECL.

`INGRES_DATE_ISO4` ([entier](#))

Équivalent à la configuration `II_DATE_FORMAT` de ISO4. Disponible depuis la version 1.2.0 de l'extension PECL.

`INGRES_DATE_GERMAN` ([entier](#))

Équivalent à la configuration `II_DATE_FORMAT` de GERMAN. Disponible depuis la version 1.2.0 de l'extension PECL.

`INGRES_DATE_MDY` ([entier](#))

Équivalent à la configuration `II_DATE_FORMAT` de MDY. Disponible depuis la version 1.2.0 de l'extension PECL.

`INGRES_DATE_DMY` ([entier](#))

Équivalent à la configuration `II_DATE_FORMAT` de DMY. Disponible depuis la version 1.2.0 de l'extension PECL.

`INGRES_DATE_YMD` ([entier](#))

Équivalent à la configuration `II_DATE_FORMAT` de YMD. Disponible depuis la version 1.2.0 de l'extension PECL.

`INGRES_MONEY_LEADING` ([entier](#))

Spécifie le caractère de devise qui doit être placé au début des valeurs monétaires.

Équivalent à la configuration `II_MONEY_FORMAT` à 'L:'. Disponible depuis la version 1.2.0 de l'extension PECL.

`INGRES_MONEY_TRAILING` ([entier](#))

Spécifie le caractère de devise qui doit être placé à la fin des valeurs monétaires.

Équivalent à la configuration `II_MONEY_FORMAT` à 'T:'. Disponible depuis la version 1.2.0 de l'extension PECL.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [ingres_autocommit](#)
- [ingres_close](#)
- [ingres_commit](#)
- [ingres_connect](#)
- [ingres_cursor](#)
- [ingres_errno](#)
- [ingres_error](#)
- [ingres_errsqlstate](#)
- [ingres_fetch_array](#)
- [ingres_fetch_object](#)
- [ingres_fetch_row](#)
- [ingres_field_length](#)
- [ingres_field_name](#)
- [ingres_field_nullable](#)
- [ingres_field_precision](#)
- [ingres_field_scale](#)
- [ingres_field_type](#)
- [ingres_num_fields](#)
- [ingres_num_rows](#)
- [ingres_pconnect](#)
- [ingres_query](#)
- [ingres_rollback](#)

8.56.8 `ingres_close()` : Ferme une connexion à un serveur Ingres

`bool ingres_close (resource link)`

`ingres_close` retourne TRUE en cas de succès, et FALSE sinon.

`ingres_close` ferme la connexion au serveur Ingres associée au lien spécifié. Si le paramètre `link` n'est pas spécifié, le dernier lien ouvert est utilisé.

Habituellement l'appel à `ingres_close` n'est pas nécessaire, car il ne ferme pas les connexions persistantes, et toutes les connexions non-persistantes sont automatiquement fermées à la fin du script.

Voir aussi `ingres_connect` et `ingres_pconnect` .

8.56.9 `ingres_commit()` : Valide une transaction

`bool ingres_commit (resource link)`

`ingres_commit` valide la transaction ouverte, ce qui rend permanentes toutes les modifications faites sur la base de données au cours de cette transaction.

Ceci ferme la transaction. Une nouvelle transaction peut être ouverte en envoyant une requête à l'aide de `ingres_query` .

Vous pouvez aussi faire en sorte que le serveur valide automatiquement les changements après chaque requête en appelant `ingres_autocommit` avant l'ouverture d'une transaction.

Voir aussi `ingres_query` , `ingres_rollback` et `ingres_autocommit` .

8.56.10 `ingres_connect()` : Ouvre une connexion à un serveur Ingres

`resource ingres_connect (string database , string username , string password)`

`ingres_connect` retourne une ressource représentant un lien Ingres II en cas de succès, et FALSE sinon.

`ingres_connect` établit une connexion avec la base de données Ingres désignée par `database` , qui suit la syntaxe `[node_id:]dbname[/svr_class]` .

Si certains paramètres sont manquants, `ingres_connect` utilise les valeurs de `ingres.default_database` , `ingres.default_user` et `ingres.default_password` indiquées dans `php.ini` .

La connexion est fermée lorsque le script se termine ou en cas d'appel à `ingres_close` .

Toutes les autres fonctions Ingres utilisent le dernier lien ouvert comme lien par défaut, il n'est donc nécessaire de conserver la valeur de retour qu'en cas d'utilisation de plus d'un lien en même temps.

Exemple pour `ingres_connect`

```
<?php
$link = ingres_connect("mydb", "user", "pass")
      or die("Erreur de connexion");
```

```
print("Connexion réussie");
ingres_close($link);
?>
```

Exemple pour [ingres_connect](#) utilisant le lien par défaut

```
<?php
ingres_connect("madb", "user", "pass")
    or die("Erreur de connexion");
echo 'Connexion réussie';
ingres_close();
?>
```

Voir aussi [ingres_pconnect](#) et [ingres_close](#) .

8.56.11 [ingres_cursor\(\)](#) : Récupère le nom du curseur pour une ressource donnée

string [ingres_cursor](#) (*resource link*)

Retourne une chaîne de caractères contenant le nom du curseur actif. Si aucun curseur n'est actif, alors NULL est retourné.

Si une ressource link est passée à la fonction [ingres_cursor](#) , elle retourne le nom du curseur enregistré pour ce lien. Si aucun lien n'est passé alors [ingres_cursor](#) retourne le nom du curseur associé avec le lien par défaut.

Exemple avec [ingres_cursor](#)

```
<?php
$link = ingres_connect($database, $user, $password);

ingres_prepare("select * from table", $link);

$cursor_name = ingres_cursor($link);

echo $cursor_name;

?>
```

Voyez aussi [ingres_prepare](#) et [ingres_execute](#) .

8.56.12 [ingres_errno\(\)](#) : Récupère le dernier numéro d'erreur d'ingres généré

int [ingres_errno](#) (*resource link*)

Retourne un entier contenant le dernier numéro d'erreur. Si aucune erreur n'est reportée, 0 est retourné.

Si une ressource link est passée à la fonction [ingres_errno](#) , elle retourne la dernière erreur enregistrée pour ce lien. Si aucun lien n'est passé alors [ingres_errno](#) retourne la dernière erreur en utilisant le lien par défaut.

La fonction, [ingres_errno](#) , devrait toujours être appelée après avoir exécuté toute sorte de requête à la base de données. Un appel d'une autre fonction avant d'appeler [ingres_errno](#) supprimera ou

changera le message d'erreur provenant du dernier appel de fonction Ingres.

Exemple avec `ingres_errno`

```
<?php
$link = ingres_connect($database, $user, $password);

ingres_query("select * from table", $link);

$error_code = ingres_errno($link);

if ( $error_code != 0 ) {
    echo "Une erreur s'est produite - " . $error_code;
}
?>
```

Voyez aussi `ingres_error` et `ingres_errsqlstate` .

8.56.13 `ingres_error()` : Récupère un message d'erreur significatif pour la dernière erreur générée

string `ingres_error` (*resource link*)

Retourne une chaîne de caractères contenant la dernière erreur ou NULL si aucune erreur ne s'est produite.

Si une ressource `link` est passée à la fonction `ingres_error` , elle retourne la dernière erreur enregistrée pour ce lien. Si aucun lien n'est passé alors `ingres_error` retourne la dernière erreur en utilisant le lien par défaut.

La fonction, `ingres_error` , devrait toujours être appelée après avoir exécuté toute sorte de requête à la base de données. Un appel d'une autre fonction avant d'appeler `ingres_error` supprimera ou changera le message d'erreur provenant du dernier appel de fonction Ingres.

Exemple avec `ingres_error`

```
<?php
ingres_connect($database, $user, $password);

ingres_query("select * from table");

$error_text = ingres_error();

if ( is_null($error_text) ) {
    echo "Une erreur s'est produite - " . $error_text;
}
?>
```

Voyez aussi `ingres_errno` et `ingres_errsqlstate` .

8.56.14 `ingres_errsqlstate()` : Gets the last SQLSTATE error code generated

string `ingres_errsqlstate` (*resource link*)

Retourne une chaîne de caractères contenant le dernier SQLSTATE ou NULL si aucune erreur ne s'est produite.

Si une ressource link est passée à la fonction `ingres_errsqlstate`, elle retourne la dernière erreur enregistrée pour ce lien. Si aucun lien n'est passé alors `ingres_errsqlstate` retourne la dernière erreur en utilisant le lien par défaut.

La fonction, `ingres_errsqlstate`, devrait toujours être appelée après avoir exécuté toute sorte de requête à la base de données. Un appel d'une autre fonction avant d'appeler `ingres_errsqlstate` supprimera ou changera le message d'erreur provenant du dernier appel de fonction Ingres.

Exemple avec `ingres_errsqlstate`

```
<?php
ingres_connect($database, $user, $password);

ingres_query("select * from table");

$error_sqlstate = ingres_errsqlstate();

if ( is_null($error_sqlstate) ) {
    echo "Une erreur s'est produite - " . $error_sqlstate;
}
?>
```

Voyez aussi `ingres_errno` et `ingres_errsqlstate`.

8.56.15 `ingres_fetch_array()` : Récupère une ligne de résultat Ingres dans un tableau

`array ingres_fetch_array (int result_type , resource link)`

`ingres_fetch_array` renvoie un tableau correspondant à la ligne récupérée, ou FALSE s'il n'y a plus de ligne à récupérer.

Cette fonction est une version améliorée de `ingres_fetch_row`. En plus de stocker les données dans un tableau à indices numériques, elle peut aussi les enregistrer dans un tableau associatif, en utilisant les noms des champs comme indices.

Si plusieurs colonnes ont le même nom, la dernière colonne aura la priorité. Pour accéder aux autres colonnes du même nom, vous devez utiliser l'index numérique, ou faire un alias pour chaque colonne.

Exemple avec `ingres_fetch_array`

```
<?php

ingres_query("select t1.f1 as foo t2.f1 as bar from t1, t2");
$result = ingres_fetch_array();
$foo = $result["foo"];
$bar = $result["bar"];

?>
```

`result_type` peut valoir `INGRES_NUM` pour un tableau à indices numériques, `INGRES_ASSOC` pour un tableau associatif, ou `INGRES_BOTH` (défaut) pour un tableau mixte (accessible selon les deux méthodes).

Du point de vue de la rapidité, cette fonction est identique à `ingres_fetch_object`, et presque aussi rapide que `ingres_fetch_row` (la différence est insignifiante).

Exemple pour `ingres_fetch_array`

```
<?php
ingres_connect($database, $user, $password);

ingres_query("select * from table");
while ($row = ingres_fetch_array()) {
    echo $row["user_id"]; // utilisation du tableau associatif
    echo $row["fullname"];
    echo $row[1];        // utilisation du tableau à indices numériques
    echo $row[2];
}
?>
```

Voir aussi [ingres_query](#) , [ingres_num_fields](#) , [ingres_field_name](#) , [ingres_fetch_object](#) et [ingres_fetch_row](#) .

8.56.16 `ingres_fetch_object()` : Récupère une ligne de résultat dans un objet

objet `ingres_fetch_object` (*int result_type* , *resource link*)

`ingres_fetch_object` renvoie un objet correspondant à la ligne (tuple) récupérée, ou FALSE s'il n'y a plus de ligne à récupérer.

`ingres_fetch_object` est similaire à `ingres_fetch_array` , avec une différence : la valeur de retour est un objet et non un tableau. Indirectement, cela signifie qu'il n'est possible d'accéder aux données qu'avec les noms des champs, et pas avec leur numéro (les nombres ne sont pas des noms valides de propriété).

Le paramètre optionnel `result_type` est une constante qui peut prendre les valeurs `INGRES_ASSOC`, `INGRES_NUM`, et `INGRES_BOTH` (par défaut).

Du point de vue de la rapidité, cette fonction est identique à `ingres_fetch_array` , et presque aussi rapide que `ingres_fetch_row` (la différence est insignifiante).

Exemple pour `ingres_fetch_object`

```
<?php
ingres_connect($database, $user, $password);
ingres_query("select * from table");
while ($row = ingres_fetch_object()) {
    echo $row->user_id;
    echo $row->fullname;
}
?>
```

Voir aussi [ingres_query](#) , [ingres_num_fields](#) , [ingres_field_name](#) , [ingres_fetch_array](#) et [ingres_fetch_row](#) .

8.56.17 `ingres_fetch_row()` : Récupère une ligne de résultat dans un tableau énuméré

array `ingres_fetch_row` (*resource link*)

`ingres_fetch_row` renvoie un tableau correspondant à la ligne récupérée, ou FALSE s'il n'y a plus de

ligne à récupérer. La ligne est stockée dans un tableau à indices numériques, le premier champ étant à l'indice 1.

Les appels successifs à [ingres_fetch_row](#) retournent les lignes suivantes du résultat, ou FALSE s'il n'y a plus de lignes.

Exemple avec [ingres_fetch_row](#)

```
<?php
ingres_connect($database, $user, $password);

ingres_query ("select * from table");
while ($row = ingres_fetch_row()) {
    echo $row[1];
    echo $row[2];
}
?>
```

Voir aussi [ingres_num_fields](#) , [ingres_query](#) , [ingres_fetch_array](#) et [ingres_fetch_object](#) .

8.56.18 [ingres_field_length\(\)](#) : Retourne la taille d'un champ

int [ingres_field_length](#) (int *index* , *resource link*)

[ingres_field_length](#) retourne la taille d'un champ. Il s'agit du nombre d'octets utilisés par le serveur pour stocker ce champ. Pour plus d'informations, voir le Guide de l'utilisateur d'Ingres/OpenAPI - Annexe C.

index est le numéro du champ et doit être compris entre 1 et la valeur donnée par [ingres_num_fields](#) .

Voir aussi [ingres_query](#) , [ingres_fetch_array](#) , [ingres_fetch_object](#) et [ingres_fetch_row](#) .

8.56.19 [ingres_field_name\(\)](#) : Retourne le nom d'un champ dans le résultat d'une requête

string [ingres_field_name](#) (int *index* , *resource link*)

[ingres_field_name](#) retourne le nom d'un champ dans le résultat d'une requête, ou FALSE en cas d'échec.

index est le numéro du champ et doit être compris entre 1 et la valeur donnée par [ingres_num_fields](#) .

Voir aussi [ingres_query](#) , [ingres_fetch_array](#) , [ingres_fetch_object](#) et [ingres_fetch_row](#) .

8.56.20 [ingres_field_nullable\(\)](#) : Teste si un champ est annulable

bool [ingres_field_nullable](#) (int *index* , *resource link*)

[ingres_field_nullable](#) retourne TRUE si le champ peut recevoir la valeur NULL et FALSE dans le cas contraire.

index est le numéro du champ et doit être compris entre 1 et la valeur donnée par [ingres_num_fields](#) .

Voir aussi [ingres_query](#) , [ingres_fetch_array](#) , [ingres_fetch_object](#) et [ingres_fetch_row](#) .

8.56.21 [ingres_field_precision\(\)](#) : Retourne la précision d'un champ

int [ingres_field_precision](#) (int index , *resource link*)

[ingres_field_precision](#) retourne la précision d'un champ. Cette valeur est utilisée uniquement pour les types de données SQL décimal, float et money. Pour plus d'informations, voir le Guide de l'utilisateur d'Ingres/OpenAPI - Annexe C.

index est le numéro du champ et doit être compris entre 1 et la valeur donnée par [ingres_num_fields](#) .

Voir aussi [ingres_query](#) , [ingres_fetch_array](#) , [ingres_fetch_object](#) et [ingres_fetch_row](#) .

8.56.22 [ingres_field_scale\(\)](#) : Retourne l'échelle d'un champ

int [ingres_field_scale](#) (int index , *resource link*)

[ingres_field_scale](#) retourne l'échelle (scale) d'un champ. Cette valeur n'est utilisée que pour le type de données SQL décimal. Pour plus d'informations, voir le Guide de l'utilisateur d'Ingres/OpenAPI - Annexe C.

index est le numéro du champ et doit être compris entre 1 et la valeur donnée par [ingres_num_fields](#) .

Voir aussi [ingres_query](#) , [ingres_fetch_array](#) , [ingres_fetch_object](#) et [ingres_fetch_row](#) .

8.56.23 [ingres_field_type\(\)](#) : Retourne le type d'un champ dans le résultat d'une requête

string [ingres_field_type](#) (int index , *resource link*)

[ingres_field_type](#) retourne le type d'un champ dans le résultat d'une requête, ou FALSE en cas d'échec. Exemples de types renvoyés par cette fonction : "IIAPI_BYTE_TYPE", "IIAPI_CHA_TYPE", "IIAPI_DTE_TYPE", "IIAPI_FLT_TYPE", "IIAPI_INT_TYPE", "IIAPI_VCH_TYPE". Certains de ces types correspondent à plus d'un type SQL, selon la taille du champ (voir [ingres_field_length](#)). Par exemple "IIAPI_FLT_TYPE" peut être un float4 ou un float8. Pour plus d'informations, voir le Guide de l'utilisateur d'Ingres/OpenAPI - Annexe C.

index est le numéro du champ et doit être compris entre 1 et la valeur donnée par [ingres_num_fields](#) .

Voir aussi [ingres_query](#) , [ingres_fetch_array](#) , [ingres_fetch_object](#) et [ingres_fetch_row](#) .

8.56.24 `ingres_num_fields()` : Retourne le nombre de champs renvoyés par la dernière requête

int `ingres_num_fields` (*resource link*)

`ingres_num_fields` retourne le nombre de champs du résultat renvoyé par le serveur Ingres après un appel à `ingres_query` .

Voir aussi `ingres_query` , `ingres_fetch_array` , `ingres_fetch_object` et `ingres_fetch_row` .

8.56.25 `ingres_num_rows()` : Retourne le nombre de lignes affectées ou renvoyées par la dernière requête

int `ingres_num_rows` (*resource link*)

Pour les requêtes DELETE, INSERT, UPDATE `ingres_num_rows` retourne le nombre de lignes (tuples) affectées par la requête. Pour les autres requêtes, `ingres_num_rows` retourne le nombre de lignes du résultat de la requête.

Note

Cette fonction est conçue principalement pour obtenir le nombre de tuples modifiés dans la base de données. Si cette fonction est appelée avant d'utiliser `ingres_fetch_array` , `ingres_fetch_object` ou `ingres_fetch_row` , le serveur efface les données du résultat et le script ne pourra plus les obtenir.

Il faut dans ce cas récupérer les données du résultat en utilisant une de ces 3 fonctions dans une boucle jusqu'à ce qu'elle renvoie FALSE , ce qui indique qu'il n'y a plus de résultats à récupérer.

Voir aussi `ingres_query` , `ingres_fetch_array` , `ingres_fetch_object` et `ingres_fetch_row` .

8.56.26 `ingres_pconnect()` : Ouvre une connexion persistante à un serveur Ingres

resource `ingres_pconnect` (*string database* , *string username* , *string password*)

`ingres_pconnect` retourne une ressource représentant un lien (link) Ingres II en cas de succès, et FALSE sinon.

Voir `ingres_connect` pour le détail des paramètres et des exemples. Il n'y a que 2 différences entre `ingres_pconnect` et `ingres_connect` : Tout d'abord, à la connexion, la fonction cherche un lien persistant déjà ouvert avec les mêmes paramètres. Si un tel lien est trouvé, un identificateur pour ce lien est retourné au lieu d'établir une nouvelle connexion. Ensuite, la connexion vers le serveur Ingres n'est pas fermée lorsque le script se termine. En fait, le lien reste ouvert pour pouvoir être réutilisé (`ingres_close` ne ferme pas les liens établis avec `ingres_pconnect`). C'est pourquoi ce type de lien est dit 'persistant'.

Voir aussi `ingres_connect` et `ingres_close` .

8.56.27 `ingres_query()` : Envoie une requête SQL à un serveur Ingres II

`bool ingres_query (string query , resource link)`

`ingres_query` retourne TRUE en cas de succès, et FALSE sinon.

`ingres_query` envoie la requête `query` au serveur Ingres. La requête doit être valide (voir le guide de référence SQL pour Ingres).

La requête s'ajoute à la transaction en cours. S'il n'y a pas de transaction ouverte, `ingres_query` en ouvre une nouvelle. Pour fermer une transaction, vous pouvez soit appeler `ingres_commit` pour valider les changements effectués sur la base de données ou `ingres_rollback` pour les annuler. Lorsque le script se termine, toute transaction ouverte est annulée (par appel à `ingres_rollback`). Vous pouvez aussi utiliser `ingres_autocommit` avant d'ouvrir une transaction pour que chaque requête SQL soit validée immédiatement et automatiquement.

Certains types de requêtes SQL ne peuvent pas être envoyés par `ingres_query` :

- CLOSE (voir `ingres_close`).
- COMMIT (voir `ingres_commit`).
- CONNECT (voir `ingres_connect`).
- DISCONNECT (voir `ingres_close`).
- get dbevent
- PREPARE TO COMMIT
- ROLLBACK (voir `ingres_rollback`).
- savepoint
- SET AUTOCOMMIT (voir `ingres_autocommit`).
- Les requêtes relatives aux curseurs ne sont pas supportées.

Exemple avec `ingres_query`

```

<?php
ingres_connect($database, $user, $password);

ingres_query("select * from table");
while ($row = ingres_fetch_row()) {
    echo $row[1];
    echo $row[2];
}
?>

```

Voir aussi `ingres_fetch_array`, `ingres_fetch_object`, `ingres_fetch_row`, `ingres_commit`, `ingres_rollback` et `ingres_autocommit`.

8.56.28 `ingres_rollback()` : Annule une transaction

`bool ingres_rollback (resource link)`

`ingres_rollback` annule (roll back) la transaction ouverte, ce qui annule les modifications faites sur la base de données au cours de cette transaction.

Ceci ferme la transaction. Une nouvelle transaction peut être ouverte en envoyant une requête à l'aide de `ingres_query`.

Voir aussi `ingres_query`, `ingres_commit` et `ingres_autocommit`.

8.57 IRC

8.57.1 Introduction

Avec IRCG, vous pouvez rapidement traiter des flots de données XML pour des milliers d'utilisateurs simultanés. Vous pouvez l'utiliser pour mettre en place rapidement des plates-formes puissantes et extensibles, pour gérer des jeux en ligne ou des salons de discussions. IRCG dispose aussi d'un mode sans flux, où une application traite les données pour fournir des fichiers aux formats tels que cHTML (i-mode) ou WML (WAP). Ces fichiers statiques sont ensuite utilisés pour améliorer les performances du serveur web.

Jusqu'en v4, IRCG fonctionne sur ces systèmes d'exploitation :

- AIX
- FreeBSD
- HP-UX
- Irix
- Linux
- Solaris
- Tru64
- Windows

Note

Cette extension a été déplacée dans le module <u>PECL</u> et ne sera plus intégrée dans PHP à partir de PHP 5.1.0.
--

8.57.2 Installation

Les instructions d'installation détaillées sont disponibles sur le site : <http://www.schumann.cx/ircg/>. Nous vous recommandons vivement d'utiliser le script d'installation fourni.

Ce n'est pas recommandé, mais vous pouvez essayer d'ajouter le support IRCG vous même. Spécifiez le chemin vers le script de configuration IRCG, `--with-ircg-config=path/to/irc-config` et, ajoutez également `--with-ircg` à votre ligne de configuration PHP.

8.57.3 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.57.4 Types de ressources

8.57.5 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)

- [Types de ressources](#)
- [Constantes pré-définies](#)
- [ircg_channel_mode](#)
- [ircg_disconnect](#)
- [ircg_eval_ecmascript_params](#)
- [ircg_fetch_error_msg](#)
- [ircg_get_username](#)
- [ircg_html_encode](#)
- [ircg_ignore_add](#)
- [ircg_ignore_del](#)
- [ircg_invite](#)
- [ircg_is_conn_alive](#)
- [ircg_join](#)
- [ircg_kick](#)
- [ircg_list](#)
- [ircg_lookup_format_messages](#)
- [ircg_lusers](#)
- [ircg_msg](#)
- [ircg_names](#)
- [ircg_nick](#)
- [ircg_nickname_escape](#)
- [ircg_nickname_unescape](#)
- [ircg_notice](#)
- [ircg_oper](#)
- [ircg_part](#)
- [ircg_pconnect](#)
- [ircg_register_format_messages](#)
- [ircg_set_current](#)
- [ircg_set_file](#)
- [ircg_set_on_die](#)
- [ircg_topic](#)
- [ircg_who](#)
- [ircg_whois](#)

8.57.7 [ircg_disconnect\(\)](#) : Ferme la connexion avec un serveur

bool [ircg_disconnect](#) (resource connection , string reason)

[ircg_disconnect](#) ferme la connexion ouverte précédemment avec un serveur grâce à la fonction [ircg-pconnect](#) , et représentée par connection .

Voir aussi [ircg_pconnect](#) .

8.57.8 [ircg_eval_ecmascript_params\(\)](#) : Décode une liste de paramètres JS-encodés

array [ircg_eval_ecmascript_params](#) (string params)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Depuis le fichier source `ircg_eval_ecmascript_params` :

```
/*
 * State 0: Looking for ' or digit
 * State 1: Assembling parameter inside '..'
 * State 2: After escape sign: Copies single char verbatim, go to 1
 * State 3: Assembling numeric para, no quotation
 * State 4: Looking for ",", skipping whitespace
 */
```

Voir aussi [ircg_lookup_format_messages](#) .

8.57.9 `ircg_fetch_error_msg()` : Retourne l'erreur rencontrée par la dernière opération IRCG

array `ircg_fetch_error_msg` (resource connection)

`ircg_fetch_error_msg` retourne l'erreur rencontrée par la dernière opération sur la connexion `connection` .

Note

Le code d'erreur est stocké dans le premier élément du tableau et le message d'erreur dans le second. Le code d'erreur est équivalent à ceux définis par la RFC 2812.

Exemple avec `ircg_fetch_error_msg`

```
<?php
if (!ircg_join ($id, "#php")) {
    $error = ircg_fetch_error_msg($id);
    echo "Can't join channel #php. Error code:
        $error[0] Description: $error[1]";
}
?>
```

8.57.10 `ircg_get_username()` : Retourne le nom d'utilisateur pour la connexion

string `ircg_get_username` (resource connection)

`ircg_get_username` retourne le nom d'utilisateur pour la connexion `connection` . Retourne FALSE si la connexion `connection` est terminée ou est invalide.

8.57.11 `ircg_html_encode()` : Prépare des données IRC pour l'affichage HTML

string `ircg_html_encode` (string html_string , bool auto_links , bool conv_br)

`ircg_html_encode` encode la chaîne HTML `html_string` pour l'affichage. Cela impose à l'interface interne d'IRCG de reformater les données reçues du serveur IRC. Cette fonction encodera les balises de formatage comme les couleurs ou les fonts mais protège certaines entités.

Voir aussi

8.57.12 ircg_ignore_add() : Ajoute un utilisateur sur la liste des utilisateurs indésirables

void **ircg_ignore_add** (resource connection , string nick)

ircg_ignore_add ajoute l'utilisateur nick dans votre liste d'utilisateurs indésirables, sur le serveur connection . Tous les messages qui vous sont envoyés par cet utilisateur seront ignorés.

Voir aussi ircg_ignore_del .

8.57.13 ircg_ignore_del() : Supprime un utilisateur de la liste des utilisateurs indésirables

bool **ircg_ignore_del** (resource connection , string nick)

ircg_ignore_del supprime l'utilisateur nick de la liste des utilisateurs indésirables, sur le serveur connection .

Voir aussi ircg_ignore_add .

8.57.14 ircg_invite() : Invite un utilisateur à rejoindre un canal

bool **ircg_invite** (resource connection , string channel , string nickname)

ircg_invite invitera l'utilisateur nickname à rejoindre le canal channel . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.57.15 ircg_is_conn_alive() : Vérifie l'état de la connexion

bool **ircg_is_conn_alive** (resource connection)

ircg_is_conn_alive retourne TRUE si la connexion connection est toujours active et fonctionnelle, et FALSE si elle n'est plus disponible.

8.57.16 ircg_join() : Rejoint un canal IRC

bool **ircg_join** (resource connection , string channel , string key)

ircg_join rejoint le canal channel sur le serveur représenté par connection . IRCG peut optionnellement passer la clé key du canal.

8.57.17 ircg_kick() : Expulse un utilisateur d'un canal

bool **ircg_kick** (resource connection , string channel , string nick , string reason)

`ircg_kick` expulse l'utilisateur nick du canal channel sur le serveur connection . Le paramètre reason doit contenir une brève explication de la raison de cette expulsion.

8.57.18 `ircg_list()` : Liste le sujet et le nombre d'utilisateurs d'un ou plusieurs canaux

bool `ircg_list` (resource connection , string channel)

`ircg_list` demandera le topic et le nombre d'utilisateurs d'un ou plusieurs canaux. La réponse est envoyée à la sortie définie par `ircg_set_file` ou `ircg_set_current` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec `ircg_list`

```
<?php
// Connexion au serveur
$id = ircg_pconnect($nickname, $ip, $port);

// définition d'un fichier pour l'affichage
ircg_set_file($id, 'irc_output.html');

// essai de rejoindre un canal
if (!ircg_join($id, $channel)) {
    echo "Echec de /join $channel<br />";
}

// envoie une commande list
ircg_list($id, $channel);

// attente pour l'affichage
sleep(5);

// déconnexion
ircg_disconnect($id, 'Au revoir le monde !');

// affichage de n'importe quoi
readfile('irc_output.html');

?>
```

L'exemple ci-dessus va afficher :

```
...
Channel #channel has n users and the topic is 'Topic'
End of LIST
...
```

Voir aussi `ircg_set_file` , `ircg_set_current` , et `ircg_who` .

8.57.19 `ircg_lookup_format_messages()` : Sélectionne un format d'affichage pour les messages IRC

bool `ircg_lookup_format_messages` (string name)

[ircg_lookup_format_messages](#) sélectionne un format d'affichage pour les messages et les événements. Les formats peuvent être prédéfinis avec la fonction [ircg_register_format_messages](#). Un format par défaut, appelé `ircg`, est toujours disponible.

Voir aussi [ircg_register_format_messages](#).

8.57.20 [ircg_lusers\(\)](#) : Statistiques réseau IRC

bool [ircg_lusers](#) (resource connection)

[ircg_lusers](#) soumet une requête de statistiques du nombre d'utilisateurs connectés au réseau via la connexion `connection`. La réponse est envoyée à la sortie définie par [ircg_set_file](#) ou [ircg_set_current](#). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [ircg_set_file](#) et [ircg_set_current](#).

8.57.21 [ircg_msg\(\)](#) : Envoie un message à un canal ou un utilisateur

bool [ircg_msg](#) (resource connection , string recipient , string message , bool suppress)

[ircg_msg](#) envoie le message `message` à l'utilisateur ou au canal sur le serveur `connection`. Un destinataire `recipient` commençant par # ou & représente un canal IRC, sinon, un utilisateur.

En donnant la valeur TRUE au paramètre `suppress`, vous éviterez que vos propres messages soient affichés sur votre connexion `connection`.

8.57.22 [ircg_names\(\)](#) : Récupère les noms d'utilisateurs visibles

bool [ircg_names](#) (int connection , string channel , string target)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Voir aussi [ircg_get_username](#) et [ircg_lusers](#).

8.57.23 [ircg_nick\(\)](#) : Change de nom sur le serveur

bool [ircg_nick](#) (resource connection , string nick)

[ircg_nick](#) change le nom (nick) que vous portez sur la connexion `connection`, si ce nom n'est pas pris.

[ircg_nick](#) retourne TRUE en cas de succès, et FALSE sinon.

8.57.24 [ircg_nickname_escape\(\)](#) : Encode les caractères spéciaux dans un pseudo pour être compatible avec IRC

string [ircg_nickname_escape](#) (string nick)

`ircg_nickname_escape` retourne le pseudo nick après l'avoir encodé pour le rendre compatible avec IRC.

Voir aussi `ircg_nickname_unescape` .

8.57.25 `ircg_nickname_unescape()` : Décode un pseudo encodé

string `ircg_nickname_unescape` (string nick)

`ircg_nickname_unescape` retourne le pseudo nick après l'avoir décodé.

Voir aussi `ircg_nickname_escape` .

8.57.26 `ircg_notice()` : Envoie une note (notice) à un utilisateur

bool `ircg_notice` (resource connection , string recipient , string message)

`ircg_notice` envoie le message message à l'utilisateur nick sur le serveur connection . Consultez votre documentation IRC pour connaître la différence exacte entre un message MSG et une note NOTICE.

8.57.27 `ircg_oper()` : Elève les privilèges au rang d'opérateur IRC

bool `ircg_oper` (resource connection , string name , string password)

`ircg_oper` identifie un utilisateur enregistré pour connection comme un opérateur IRC. Les paramètres name et password doivent correspondre à un compte d'opérateur IRC. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.57.28 `ircg_part()` : Quitte le canal

bool `ircg_part` (resource connection , string channel)

`ircg_part` quitte le canal channel sur le serveur représenté par connection .

8.57.29 `ircg_pconnect()` : Connecte à un serveur IRC

resource `ircg_pconnect` (string username , string server_ip , int server_port , string msg_format , array ctcp_messages , array user_settings , bool bailout_on_trivial)

`ircg_pconnect` essaie d'établir une connexion avec le serveur IRC server_ip , et retourne une ressource de connexion pour utilisation ultérieure.

Le seul paramètre obligatoire est username , qui représente le nom d'utilisateur IRC initial. server_ip et server_port sont optionnels et, par défaut, valent respectivement 127.0.0.1 (hôte local) et 6667 .

Note

Actuellement, le paramètre `server_ip` n'effectue aucune résolution de nom et n'accepte que les IP au format numérique.

Vous pouvez personnaliser l'affichage des messages IRC et les événements qui s'y rattachent avec les formats de messages générés par la fonction [ircg_register_format_messages](#), en spécifiant le format dans `msg_format`.

Si vous voulez traiter les messages CTCP comme des messages ACTION (/me), vous devez définir un type CTCP (e.g. ACTION) en une chaîne personnalisée. Faîte le en passant un tableau associatif au paramètre `ctcp_messages`. Les clés de ce tableau sont de type CTCP et les valeurs respectives sont dans un message formaté.

Vous pouvez définir les marques " ident ", " password ", et " realname " qui sont envoyées au serveur IRC en les définissant dans un tableau associatif passé au paramètre `user_settings`.

Voir aussi [ircg_disconnect](#), [ircg_is_conn_alive](#) et [ircg_register_format_messages](#).

8.57.30 [ircg_register_format_messages\(\)](#) : Enregistre un nouveau format d'affichage des messages IRC

`bool ircg_register_format_messages (string name , array messages)`

[ircg_register_format_messages](#) vous permet de personnaliser l'affichage de vos messages IRC. Vous pouvez même enregistrer plusieurs formats et passer de l'un à l'autre à la volée avec [ircg_lookup_format_messages](#).

- Message public brut
 - Message privé reçu
 - Message privé envoyé
 - Un utilisateur quitte le canal
 - Un utilisateur rejoint le canal
 - Un utilisateur est expulsé du canal
 - Le sujet du canal est modifié
 - Erreur
 - Erreur fatale
 - Rejoint la fin de la liste (??? : Join list end)
 - Se quitte soi-même (??? : Self part)
 - Un utilisateur s'expulse lui-même
 - Un utilisateur quitte sa connexion
 - Début de regroupement en masse
 - Élément de regroupement en masse
 - Fin de regroupement en masse
 - Whois utilisateur
 - Whois serveur
 - Whois inactif
 - Whois canal
 - Fin de whois
 - Changement de statut Voice pour un utilisateur
 - Changement de statut d'opérateur pour un utilisateur
 - Liste d'utilisateurs indésirables
 - Fin de liste d'utilisateurs indésirables
-
- %f - origine
 - %t - destination

- %c - canal
 - %r - message brut
 - %m - message encodé
 - %j - message encodé js
-
- 1 - encodage mod
 - 2 - nickname decode

Voir aussi [ircg_lookup_format_messages](#) .

8.57.31 `ircg_set_current()` : Prépare la connexion courante pour l'affichage

`bool ircg_set_current (resource connection)`

`ircg_set_current` sélectionne la connexion courante pour l'affichage dans le contexte d'exécution courant. Tous les messages envoyés par le serveur représenté par `connection` seront recopiés et envoyés à la sortie standard, avec le format standard, ou bien la chaîne de format spécifiée par la fonction [ircg_register_format_messages](#) .

Voir aussi [ircg_register_format_messages](#) .

8.57.32 `ircg_set_file()` : Configure le journal de connexion

`bool ircg_set_file (resource connection , string path)`

`ircg_set_file` spécifie un journal path dans lequel toutes les sorties de la connexion `connection` seront écrites. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.57.33 `ircg_set_on_die()` : Configure une action à exécuter lorsque la connexion IRC se termine

`bool ircg_set_on_die (resource connection , string host , int port , string data)`

Au cas où la connexion `connection` se termine, IRCG se connectera à `host` sur le port `port` (note : `host` doit être une adresse IPv4, car IRCG ne résout pas les noms d'hôtes), enverra `data` à la nouvelle connexion et attendra que la connexion distante soit fermée par le serveur. Cela peut être utile pour appeler un script PHP par exemple.

Cette fonctionnalité requiert IRCG 3.

8.57.34 `ircg_topic()` : Modifie le sujet (topic) d'un canal

`bool ircg_topic (resource connection , string channel , string new_topic)`

`ircg_topic` change le sujet du canal de `channel` en `new_topic` , sur le serveur `connection` .

8.57.35 `ircg_who()` : Interroge le serveur sur les informations WHO

`bool ircg_who (resource connection , string mask , bool ops_only)`

`ircg_who` demandera une liste d'utilisateurs dont le pseudo correspond au masque `mask` sur le réseau `connection` . Le paramètre optionnel `ops_only` réduira la liste aux seuls opérateurs du serveur.

La réponse est envoyée vers la sortie définie par `ircg_set_file` ou `ircg_set_current` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `ircg_set_file` , et `ircg_set_current` .

8.57.36 `ircg_whois()` : Requier les informations sur un utilisateur

`bool ircg_whois (resource connection , string nick)`

`ircg_whois` envoie une requête au serveur `connection` pour lui demander des informations sur l'utilisateur `nick` .

8.58 Intégration de Java avec PHP

8.58.1 Introduction

Il y a deux moyens de connecter PHP et Java : soit en intégrant PHP dans un environnement de Servlet Java , ce qui est la solution la plus stable et la plus efficace, soit en intégrant Java directement dans PHP. La première solution est fournie par le module SAPI qui s'interface avec un serveur de Servlet, la dernière par cette extension.

L'extension Java fournit un moyen simple et efficace pour créer et invoquer des méthodes d'un objet Java depuis PHP. La JVM est créée avec JNI, et tout fonctionne en intra-processus.

Attention

Ce module est **EXPERIMENTAL** . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez ce module à vos risques et périls.

8.58.2 Pré-requis

Vous avez besoin d'une Machine virtuelle Java installée sur votre machine pour utiliser cette extension.

8.58.3 Installation

Cette extension PECL n'est pas intégrée à PHP.

En PHP 4, les sources de cette extension PECL peuvent être trouvées dans le dossier ext/ avec les sources de PHP ou sur le lien PECL ci-dessous. Si vous voulez utiliser ces fonctions, vous devez compiler PHP avec le support Java en utilisant l'option de configuration --with-java[=DIR] où DIR représente le dossier d'installation de base de votre JDK. Cette extension ne peut être compiler qu'en tant qu'extension partagée. Les extensions compilées aditionnellement peuvent être trouvées dans php-src/ext/java/README .

Les utilisateurs de Windows doivent activer la bibliothèque php_java.dll dans le php.ini pour pouvoir utiliser ces fonctions. En PHP 4, cette bibliothèque DLL se trouve dans le dossier extensions/ avec les binaires PHP pour Windows téléchargées. Vous pouvez télécharger ces bibliothèques DLL pour ces extensions PECL depuis la page de téléchargement de PHP sous sur la page <http://snaps.php.net/> .

Note

Pour activer ce module dans un environnement Windows avec PHP <= 4.0.6, vous devez rendre la bibliothèque jvm.dll disponible pour votre PATH système. Aucune autre bibliothèque DLL n'est nécessaire pour les version de PHP > 4.0.6.

8.58.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

Nom	Par défaut	Modifiable	Historique
java.class.path	NULL	PHP_INI_ALL	

java.home	NULL	PHP_INI_ALL
java.library.path	NULL	PHP_INI_ALL
java.library	JAVALIB	PHP_INI_ALL

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [Directives du php.ini](#) .

8.58.5 Types de ressources

Cette extension ne définit aucune ressource.

8.58.6 Constantes pré-définies

Cette extension ne définit aucune constante.

8.58.7 Exemples

Exemple avec Java

```
<?php
// créer une instance de la classe Java java.lang.System dans PHP
$system = new Java('java.lang.System');

// accède aux propriétés
echo 'Java version=' . $system->getProperty('java.version') . ' <br />';
echo 'Java vendor=' . $system->getProperty('java.vendor') . ' <br />';
echo 'OS=' . $system->getProperty('os.name') . ' ' .
      $system->getProperty('os.version') . ' on ' .
      $system->getProperty('os.arch') . ' <br />';

// Exemple avec java.util.Date
$formater = new Java('java.text.SimpleDateFormat',
                    "EEEE, MMMM dd, yyyy 'at' h:mm:ss a zzzz");

echo $formater->format(new Java('java.util.Date'));
?>
```

Exemple AWT

```
<?php
// Cet exemple ne fonctionne qu'en mode CGI.

$frame = new Java('java.awt.Frame', 'PHP');
$button = new Java('java.awt.Button', 'Hello Java World!');

$frame->add('North', $button);
$frame->validate();
$frame->pack();
$frame->visible = True;

$thread = new Java('java.lang.Thread');
$thread->sleep(10000);

$frame->dispose();
?>
```

Notes :

- new Java() crée une nouvelle instance d'une classe, si un constructeur valable est disponible. Si aucun paramètre n'est passé, et le constructeur par défaut est utile pour

accéder à ces classes telles que "java.lang.System", qui fournissent leur fonctionnalités via des méthodes statiques.

- Lors de l'accès aux membres d'une instance, PHP commencera par rechercher les membres Bean, puis les champs publics. En d'autres termes, " print \$date.time " sera d'abord résolu par " \$date.getTime() ", puis par " \$date.time ";
- Les membres statiques et d'instance sont accessibles avec la même syntaxe. De plus, si un objet est de type " java.lang.Class ", les membres statiques de la classe (champs et méthodes) sont accessibles.
- Les exceptions sont transformées en alertes PHP, et résultat NULL. Les alertes peuvent être supprimées en préfixant l'appel par l'opérateur > . Les fonctions suivantes peuvent être utilisées pour lire et effacer la dernière erreur remontée :
 - ◆ [java_last_exception_get](#)
 - ◆ [java_last_exception_clear](#)
- Les surcharges de fonctions sont des problèmes épineux, étant données les différences de type de valeurs entre les deux /bin/bash: q: command not found mais efficace pour déterminer la meilleur fonction à utiliser.

De plus, les noms de méthodes ne sont pas sensibles à la casse en PHP, ce qui augmente le nombre de conflits potentiels.

Une fois qu'une méthode est sélectionnée, les paramètres sont transtypés, avec une perte d'information potentielle non négligeable (par exemple, les nombres à virgules flottante en double précisions seront convertis en booléen).

- Traditionnellement en PHP, les tableaux et les tables de hashage peuvent être interchangeables, et fonctionnent de la même façon. Notez que les tables de hashage de PHP ne peuvent être indexées qu'avec des entiers ou des chaînes, et que le type primitif de tableau de Java ne peut comporter de trous dans les index. Notez aussi que les valeurs sont passées par valeur, ce qui peut être coûteux en mémoire et en temps.

8.58.8 Java Servlet SAPI

L'interface PHP 4 sapi/servlet est construite sur un mécanisme défini par l'extension Java, qui permet à PHP d'être exécuté comme une servlet. L'avantage immédiat d'un point de vue PHP est que les serveurs web qui supportent les servlets gèrent rigoureusement les machines virtuelles. Les instructions d'installation du module Servlet SAPI sont disponibles dans le fichier php4/sapi/README . Notes :

- Bien que ce code soit prévu pour fonctionner sur n'importe quel serveur à Servlet, il n'a été testé qu'avec le module Apache Jakarta/tomcat (jusqu'à aujourd'hui). Les remontées de bogues, les réussites et les patches nécessaires pour faire fonctionner ce code sur d'autres serveurs sont fortement encouragés !
 - PHP a l'habitude de changer le dossier de travail. Le serveur SAPI/Servlet le changera à nouveau, mais tant que PHP fonctionnera, le moteur de servlet ne pourra pas charger de classes dans le CLASSPATH, si le dossier est spécifié avec un chemin relatif, ou ne pourra pas trouver le dossier d'administration et de compilation des tâches JSP.
- [Introduction](#)
 - [Pré-requis](#)
 - [Installation](#)
 - [Configuration à l'exécution](#)
 - [Types de ressources](#)
 - [Constantes pré-définies](#)
 - [Exemples](#)

- [Java Servlet SAPI](#)
- [java_last_exception_clear](#)
- [java_last_exception_get](#)

8.58.10 java_last_exception_get() : Lit la dernière exception Java

object `java_last_exception_get` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

L'exemple ci-dessous montre l'utilisation du gestionnaire d'exceptions java depuis PHP :

Gestionnaire d'exceptions Java

```
<?php
$stack = new Java('java.util.Stack');
$stack->push(1);

// Cela doit marcher
$result = $stack->pop();
$ex = java_last_exception_get();
if (!$ex) {
    echo "$result\n";
}

// Cela doit échouer (le rapport d'erreurs est supprimé par @)
$result = @$stack->pop();
$ex = java_last_exception_get();
if ($ex) {
    echo $ex->toString();
}

// Efface la dernière exception
java_last_exception_clear();
?>
```

8.59 KADM5

8.59.1 Introduction

Ce paquet vous permet d'accéder à l'administration des serveurs Kerberos V. Vous pourrez créer, modifier et effacer les directives et les éléments principaux Kerberos V.

Plus d'informations à propos de Kerberos peuvent être trouvées sur <http://web.mit.edu/kerberos/www/> .

La documentation sur Kerberos et KADM5 peut être trouvée sur http://web.mit.edu/kerberos/www/krb5-1.2/krb5-1.2.8/doc/admin_toc.html .

8.59.2 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

8.59.2.1 Constantes pour les attributs des balises

Les fonctions `kadm5_create_principal` , `kadm5_modify_principal` et `kadm5_modify_principal` vous permettent de spécifier des attributs spéciaux utilisant un champ de bits. Les symboles sont définis ci-dessous :

constante
KRB5_KDB_DISALLOW_POSTDATED
KRB5_KDB_DISALLOW_FORWARDABLE
KRB5_KDB_DISALLOW_TGT_BASED
KRB5_KDB_DISALLOW_RENEWABLE
KRB5_KDB_DISALLOW_PROXIABLE
KRB5_KDB_DISALLOW_DUP_SKEY
KRB5_KDB_DISALLOW_ALL_TIX
KRB5_KDB_REQUIRES_PRE_AUTH
KRB5_KDB_REQUIRES_HW_AUTH
KRB5_KDB_REQUIRES_PWCHANGE
KRB5_KDB_DISALLOW_SVR
KRB5_KDB_PWCHANGE_SERVER
KRB5_KDB_SUPPORT_DESMD5
KRB5_KDB_NEW_PRINC

8.59.2.2 Constantes pour les options

Les fonctions `kadm5_create_principal` , `kadm5_modify_principal` et `kadm5_get_principal` permettent de spécifier ou retourner les options des éléments principaux en tant que tableaux associatifs. Les clés pour le tableau associatif sont définies en tant que constantes ci-dessous :

constante	funcdef	description
-----------	---------	-------------

KADM5_PRINCIPAL	long	Le délai d'expiration des éléments principaux en tant que timestamp Kerberos.
KADM5_PRINC_EXPIRE_TIME	long	Le délai d'expiration des éléments principaux en tant que timestamp Kerberos.
KADM5_LAST_PW_CHANGE	long	Le délai pendant lequel le mot de passe du principal a été changé pour la dernière fois.
KADM5_PW_EXPIRATION	long	Le délai d'expiration du mot de passe du principal courant, en tant que timestamp Kerberos.
KADM5_MAX_LIFE	long	La durée de vie maximale de tous les tickets Kerberos issus de ce principal.
KADM5_MAX_RLIFE	long	La durée maximale renouvelable de tous les tickets Kerberos issus de ce principal.
KADM5_MOD_NAME	string	Le nom du principal Kerberos qui a modifié ce principal le plus récemment.
KADM5_MOD_TIME	long	L'heure à laquelle ce principal a été modifié pour la dernière fois, en tant que timestamp Kerberos.
KADM5_KVNO	long	La version de la clé courante du principal.
KADM5_POLICY	string	Le nom de la directive contrôlant ce principal.
KADM5_CLEARPOLICY	long	La procédure standart servant à assigner la directive par défaut pour créer des éléments principaux. KADM5_CLEARPOLICY supprime ce comportement.
KADM5_LAST_SUCCESS	long	L'heure KDC de la dernière AS_REQ qui a réussi.
KADM5_LAST_FAILED	long	L'heure KDC de la dernière AS_REQ qui a échoué.
KADM5_FAIL_AUTH_COUNT	long	Le nombre d'échec récursif AS_REQs.
KADM5_RANDKEY	long	Génère un mot de passe aléatoire pour le principal. Le paramètre password sera ignoré.
KADM5_ATTRIBUTES	long	Un champ de bites d'attributs à utiliser par KDC.

8.59.3 Exemples

Cet exemple simple montre comment se connecter, interroger, imprimer les éléments principaux et se déconnecter depuis une base de données KADM5.

Exemple général de l'extension KADM5
<pre><?php \$handle = kadm5_init_with_password("afs-1", "GONICUS.LOCAL", "admin/admin", "password"); print "<h1>get_principals</h1>\n"; \$principals = kadm5_get_principals(\$handle); for(\$i=0; \$i<count(\$principals); \$i++) print "\$principals[\$i]
\n"; print "<h1>get_policies</h1>\n"; \$policies = kadm5_get_policies(\$handle); for(\$i=0; \$i<count(\$policies); \$i++) print "\$policies[\$i]
\n"; print "<h1>get_principal burbach@GONICUS.LOCAL</h1>\n"; \$options = kadm5_get_principal(\$handle, "burbach@GONICUS.LOCAL"); \$keys = array_keys(\$options); for(\$i=0; \$i<count(\$keys); \$i++) { \$value = \$options[\$keys[\$i]]; } }</pre>

```

    print "$keys[$i]: $value<br>\n";
}

$options = array(KADM5_PRINC_EXPIRE_TIME => 0);
kadm5_modify_principal($handle, "burbach@GONICUS.LOCAL", $options);

kadm5_destroy($handle);
?>

```

8.59.4 Contacts

Si vous avez des commentaires, des corrections de bogues, des enrichissements ou si vous voulez aider dans le développement, vous pouvez envoyer un email à holger.burbach@gonicus.de . La page du projet peut être trouvé sur http://oss.gonicus.de/project/?group_id=7 .

- [Introduction](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [Contacts](#)
- [kadm5_chpass_principal](#)
- [kadm5_create_principal](#)
- [kadm5_delete_principal](#)
- [kadm5_destroy](#)
- [kadm5_flush](#)
- [kadm5_get_policies](#)
- [kadm5_get_principal](#)
- [kadm5_get_principals](#)
- [kadm5_init_with_password](#)
- [kadm5_modify_principal](#)

8.59.6 kadm5_create_principal() : Crée un principal kerberos avec les paramètres donnés

bool **kadm5_create_principal** (resource handle , string principal , *string password* , array options)

[kadm5_create_principal](#) crée un principal avec le mot de passe password donné. Si password n'est pas spécifié ou s'il vaut NULL , une clé aléatoire sera générée.

Il est possible de spécifier plusieurs paramètres optionnels avec le tableau options . Les options suivantes sont autorisées : KADM5_PRINC_EXPIRE_TIME, KADM5_PW_EXPIRATION, KADM5_ATTRIBUTES, KADM5_MAX_LIFE, KADM5_KVNO, KADM5_POLICY, KADM5_CLEARPOLICY, KADM5_MAX_RLIFE.

8.59.7 kadm5_delete_principal() : Efface un principal kerberos

bool **kadm5_delete_principal** (resource handle , string principal)

[kadm5_delete_principal](#) efface un principal depuis la base de données Kerberos.

8.59.8 kadm5_destroy() : Ferme la connexion avec le serveur d'administration et libère toutes les ressources associées

bool **kadm5_destroy** (resource handle)

8.59.9 kadm5_flush() : Valide toutes les modifications de la base de données Kerberos, et quitte la connexion ouverte du serveur d'administration Kerberos

bool **kadm5_flush** (resource handle)

8.59.10 kadm5_get_policies() : Récupère toutes les directives depuis la base de données Kerberos

array **kadm5_get_policies** (resource handle)

kadm5_get_policies retourne un tableau contenant tous les noms des directives.

8.59.11 kadm5_get_principal() : Récupère les entrées des éléments principaux depuis la base de données Kerberos

array **kadm5_get_principal** (resource handle , string principal)

kadm5_get_principal retourne un tableau associatif contenant les clés suivantes : KADM5_PRINCIPAL, KADM5_PRINC_EXPIRE_TIME, KADM5_PW_EXPIRATION, KADM5_ATTRIBUTES, KADM5_MAX_LIFE, KADM5_MOD_NAME, KADM5_MOD_TIME, KADM5_KVNO, KADM5_POLICY, KADM5_MAX_RLIFE, KADM5_LAST_SUCCESS, KADM5_LAST_FAILED, KADM5_FAIL_AUTH_COUNT.

8.59.12 kadm5_get_principals() : Récupère tous les éléments principaux depuis la base de données Kerberos

array **kadm5_get_principals** (resource handle)

kadm5_get_principals retourne un tableau contenant tous les noms des éléments principaux.

8.59.13 kadm5_init_with_password() : Ouvre une connexion à la bibliothèque KADM5 et initialise toutes les informations nécessaires

resource **kadm5_init_with_password** (string admin_server , string realm , string principal , string password)

kadm5_init_with_password ouvre une connexion avec la bibliothèque KADM5 en utilisant le principal et le mot de passe password pour obtenir les créances initiales depuis le serveur d'administration admin_server . Le paramètre realm définit le domaine d'identification pour la connexion.

8.59.14 kadm5_modify_principal() : Modifie un principal Kerberos avec les paramètres donnés

bool **kadm5_modify_principal** (resource handle , string principal , array options)

kadm5_modify_principal modifie un principal en utilisant les options options . Les constantes suivantes sont autorisées : KADM5_PRINC_EXPIRE_TIME, KADM5_PW_EXPIRATION, KADM5_ATTRIBUTES, KADM5_MAX_LIFE, KADM5_KVNO, KADM5_POLICY, KADM5_CLEARPOLICY, KADM5_MAX_RLIFE, KADM5_FAIL_AUTH_COUNT.

8.60 LDAP

8.60.1 Introduction

LDAP signifie : Lightweight Directory Access Protocol (Protocole léger d'accès aux annuaires). C'est un protocole utilisé pour accéder aux "serveurs de dossiers". Ces serveurs sont des bases de données particulières, qui stockent les informations sous forme d'arborescence.

Le concept d'arborescence est similaire à celui de la structure de votre système de fichier, hormis le fait que dans ce contexte, la racine s'appelle "le monde", et que le premier niveau de sous-dossier s'appelle "pays". Les niveaux encore en dessous sont des "compagnies" "organisation" ou "places", et encore plus bas, vous trouverez des "personnes" et même des "équipements" et "documents".

Pour identifier un fichier dans votre disque, vous utilisez un chemin tel que

```
/usr/local/mon_application/documents
```

Le slash indique une division dans la référence, et la séquence est lue de gauche à droite.

l'équivalent d'une référence globale en LDAP s'appelle un "nom distingué" ("distinguished name"), aussi appelé "dn". Un exemple de dn serait :

```
cn=Jean Dupond,ou=Comptabilité,o=Ma Compagnie,c=FR
```

La virgule marque la séparation de chaque division comme référence, et la séquence est lue de droite à gauche. Il faut donc lire :

```
country = FR  
organization = Ma Compagnie  
organizationalUnit = Comptabilité  
commonName = Jean Dupond
```

De la même façon qu'il n'y a pas de règle obligatoire sur comment organiser les fichiers sur un disque dur, un responsable de serveur de dossiers peut organiser le serveur comme cela lui semble le plus pratique. Cependant, il y a des conventions à utiliser. Le principe est que vous ne pouvez pas accéder à un serveur de dossier à moins que vous ne connaissiez sa structure, de même que vous ne pouvez écrire une base de données sans en connaître les tables et les bases.

Bien plus d'informations sont disponibles aux URL suivantes (en anglais) :

- [Mozilla](#)
- [OpenLDAP Project](#)
- [LDAP World](#)

Le SDK Netscape contient un guide du programmeur ([Programmer's Guide](#)) très utile, au format HTML (et en anglais).

8.60.2 Pré-requis

Vous devez télécharger et compiler les bibliothèques clientes LDAP depuis soit [OpenLDAP](#) ou [Bind9.net](#) pour assurer le support LDAP.

8.60.3 Installation

Le support LDAP de PHP n'est pas activé par défaut. Vous devez utiliser l'option de configuration `--with-ldap[=DIR]` lorsque vous compilez PHP, où DIR est le répertoire d'installation du serveur LDAP. Pour activer le support SASL, assurez-vous que l'option de configuration `--with-ldap-sasl[=DIR]` est utilisée et que le fichier `sasl.h` existe sur votre système.

Note

Note aux utilisateurs Win32

Afin d'activer ce module dans l'environnement Windows, vous devez copier plusieurs bibliothèques depuis le dossier DLL de PHP/Win32 dans le dossier système SYSTEM32 de votre machine (par exemple : `C:\WINNT\SYSTEM32` ou `C:\WINDOWS\SYSTEM32`). Pour PHP `<= 4.2.0`, copiez `libsasl.dll` , pour PHP `>= 4.3.0`, copiez `libeay32.dll` et `ssleay32.dll` dans votre dossier système SYSTEM.

Pour pouvoir utiliser les bibliothèques Oracle LDAP, un [environnement Oracle](#) propre doit être défini.

8.60.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable	Historique
<code>ldap.max_links</code>	"-1"	PHP_INI_SYSTEM	

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#) .

8.60.5 Types de ressources

Cette extension ne définit aucune ressource.

8.60.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

`LDAP_DEREF_NEVER` ([entier](#))
`LDAP_DEREF_SEARCHING` ([entier](#))
`LDAP_DEREF_FINDING` ([entier](#))
`LDAP_DEREF_ALWAYS` ([entier](#))
`LDAP_OPT_DEREF` ([entier](#))
`LDAP_OPT_SIZELIMIT` ([entier](#))
`LDAP_OPT_TIMELIMIT` ([entier](#))
`LDAP_OPT_PROTOCOL_VERSION` ([entier](#))

LDAP_OPT_ERROR_NUMBER (entier)
 LDAP_OPT_REFERRALS (entier)
 LDAP_OPT_RESTART (entier)
 LDAP_OPT_HOST_NAME (entier)
 LDAP_OPT_ERROR_STRING (entier)
 LDAP_OPT_MATCHED_DN (entier)
 LDAP_OPT_SERVER_CONTROLS (entier)
 LDAP_OPT_CLIENT_CONTROLS (entier)
 LDAP_OPT_DEBUG_LEVEL (entier)
 GSLC_SSL_NO_AUTH (entier)
 GSLC_SSL_ONEWAY_AUTH (entier)
 GSLC_SSL_TWOWAY_AUTH (entier)

8.60.7 Exemples

Lit les informations sur toutes les entrées dont le nom commence par "S" sur le serveur de dossier, puis affiche le nom et l'adresse email.

Recherche avec LDAP

```
<?php
// La séquence de base avec LDAP est
// connexion, liaison, recherche, interprétation du résultat
// déconnexion

echo '<h3>requête de test de LDAP</h3>';
echo 'Connexion ...';
$ds=ldap_connect("localhost"); // doit être un serveur LDAP valide !
echo 'Le résultat de connexion est ' . $ds . '<br />';

if ($ds) {
    echo 'Liaison ...';
    $r=ldap_bind($ds); // connexion anonyme, typique
                        // pour un accès en lecture seule.
    echo 'Le résultat de connexion est ' . $r . '<br />';

    echo 'Recherchons (sn=S*) ...';
    // Recherche par nom
    $sr=ldap_search($ds,"o=My Company, c=US", "sn=S*");
    echo 'Le résultat de la recherche est ' . $sr . '<br />';

    echo 'Le nombre d\'entrées retournées est ' . ldap_count_entries($ds,$sr)
        . '<br />';

    echo 'Lecture des entrées ...<br />';
    $info = ldap_get_entries($ds, $sr);
    echo 'Données pour ' . $info["count"] . ' entrées:<br />';

    for ($i=0; $i<$info["count"]; $i++) {
        echo 'dn est : ' . $info[$i]["dn"] . '<br />';
        echo 'premiere entree cn : ' . $info[$i]["cn"][0] . '<br />';
        echo 'premier email : ' . $info[$i]["mail"][0] . '<br />';
    }

    echo 'Fermeture de la connexion';
    ldap_close($ds);
} else {
    echo '<h4>Impossible de se connecter au serveur LDAP.</h4>';
}
?>
```

8.60.8 Utiliser les fonctions LDAP de PHP

Avant d'utiliser les fonctions LDAP, vous devez connaître

- Le nom ou l'adresse du serveur de dossiers que vous voudrez utiliser
- Le "base dn" du serveur (la partie du dossier monde qui est disponible sur ce serveur, ce qui peut être "o=Ma Compagnie,c=FR")
- Le mot de passe éventuel d'accès au serveur (de nombreux serveurs fournissent un accès anonyme en lecture, mais requièrent des mots de passe pour tout le reste).

La séquence LDAP typique que vous exécuterez sera la suivante :

```
ldap_connect()    // établit une connexion au serveur
|
ldap_bind()      // connexion anonyme ou identifiée
|
réalisation de commandes comme des recherches ou des
modifications, puis affichage du résultat.
|
ldap_close()     // déconnexion
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [Utiliser les fonctions LDAP de PHP](#)
- [ldap_8859_to_t61](#)
- [ldap_add](#)
- [ldap_bind](#)
- [ldap_close](#)
- [ldap_compare](#)
- [ldap_connect](#)
- [ldap_count_entries](#)
- [ldap_delete](#)
- [ldap_dn2ufn](#)
- [ldap_err2str](#)
- [ldap_errno](#)
- [ldap_error](#)
- [ldap_explode_dn](#)
- [ldap_first_attribute](#)
- [ldap_first_entry](#)
- [ldap_first_reference](#)
- [ldap_free_result](#)
- [ldap_get_attributes](#)
- [ldap_get_dn](#)
- [ldap_get_entries](#)
- [ldap_get_option](#)
- [ldap_get_values_len](#)
- [ldap_get_values](#)
- [ldap_list](#)
- [ldap_mod_add](#)
- [ldap_mod_del](#)

- [ldap_mod_replace](#)
- [ldap_modify](#)
- [ldap_next_attribute](#)
- [ldap_next_entry](#)
- [ldap_next_reference](#)
- [ldap_parse_reference](#)
- [ldap_parse_result](#)
- [ldap_read](#)
- [ldap_rename](#)
- [ldap_sasl_bind](#)
- [ldap_search](#)
- [ldap_set_option](#)
- [ldap_set_rebind_proc](#)
- [ldap_sort](#)
- [ldap_start_tls](#)
- [ldap_t61_to_8859](#)
- [ldap_unbind](#)

8.60.10 ldap_add() : Ajoute une entrée dans un dossier LDAP

bool **ldap_add** (resource link_identifier , string dn , array entry)

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

[ldap_add](#) sert à ajouter une entrée dans un dossier LDAP. dn est l'identification complète de l'entrée qui sera ajoutée, et entry est un tableau avec les informations sur la nouvelle entrée. Ces valeurs sont indexées individuellement. En cas de valeurs multiples pour un attribut, elle sont indexées numériquement, à partir de 0.

Exemple de tableau pour ajouter une entrée LDAP

```
<?php
$entree["attribut1"] = "valeur";
$entree["attribut2"][0] = "valeur1";
$entree["attribut2"][1] = "valeur2";
?>
```

Exemple complet avec identification LDAP

```
<?php
$ds=ldap_connect("localhost"); // on suppose que le serveur LDAP est sur le serveur local

if ($ds) {
    // Connexion avec une identité qui permet les modifications
    $r=ldap_bind($ds, "cn=root, o=Ma Compagnie, c=FR", "secret");

    // prepare les données
    $info["cn"]='Jean Dupont';
    $info["sn"]='Jean';
    $info["mail"]='jeand@serveur.mail.com';
    $info["objectclass"]='personne';

    // Ajoute les données au dossier
    $r=ldap_add($ds, "cn=Jean Dupont, o=Ma Compagnie, c=FR", $info);

    ldap_close($ds);
} else {
    echo 'Impossible de se connecter au serveur LDAP';
}
```

?>

Note

Cette fonction gère les chaînes binaires.

8.60.11 ldap_bind() : Authentification au serveur LDAP

bool **ldap_bind** (resource link_identifier , string bind_rdn , string bind_password)

`ldap_bind` s'identifie auprès du serveur LDAP `link_identifier` , avec le nom d'utilisateur `bind_rdn` et le mot de passe `bind_password` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

`ldap_bind` effectue une opération de bind avec le serveur. `bind_rdn` et `bind_password` sont optionnels. S'ils sont omis, une connexion anonyme est tentée.

Identification avec LDAP

```
<?php
// Eléments d'identification LDAP
$ldaprdn = 'nom d\'utilisateur'; // DN ou RDN LDAP
$ldappass = 'mot de passe'; // Mot de passe associé

//Connexion au serveur LDAP
$ldapconn = ldap_connect("ldap.example.com")
    or die("Impossible de se connecter au serveur LDAP.");

if ($ldapconn) {

    //Connexion au serveur LDAP
    $ldapbind = ldap_bind($ldapconn, $ldaprdn, $ldappass);

    // Identification
    if ($ldapbind) {
        echo "Connexion LDAP réussie";
    } else {
        echo "Connexion LDAP échouée";
    }
}

?>
```

Connexion anonyme à un serveur LDAP

```
<?php
//Connexion anonyme à un serveur LDAP

//Connexion au serveur LDAP
$ldapconn = ldap_connect("ldap.example.com")
    or die("Impossible de se connecter au serveur LDAP.");

if ($ldapconn) {

    // identification anonyme
    $ldapbind = ldap_bind($ldapconn);

    if ($ldapbind) {
        echo 'Connexion LDAP anonyme réussie';
    } else {
        echo 'Connexion LDAP anonyme échouée';
    }
}
```

```

    }
}
?>

```

8.60.12 `ldap_close()` : Alias de `ldap_unbind`

Cette fonction est un alias de : `ldap_unbind` .

8.60.13 `ldap_compare()` : Compare une entrée avec des valeurs d'attributs

mixed `ldap_compare` (resource link_identifier , string dn , string attribute , string value)

`ldap_compare` retourne TRUE si value correspond et, sinon, FALSE . `ldap_compare` retourne -1 en cas d'erreur.

`ldap_compare` sert à comparer la valeur value de l'attribut attribute à la valeur du même attribut de l'entrée dn .

L'exemple suivant montre comment vérifier que deux mots de passe correspondent, dont l'un est celui d'une entrée du serveur LDAP.

Exemple complet de vérification de mot de passe avec LDAP

```

<?php

$ds=ldap_connect("localhost"); // doit être un serveur LDAP valide!

if ($ds) {

    // identification
    if (ldap_bind($ds)) {

        // preparation des données
        $dn = "cn=Matti Meikku, ou=Mon Service, o=Ma Compagnie, c=FR";
        $value = "secretpassword";
        $attr = "password";

        // comparaison des valeurs
        $r=ldap_compare($ds, $dn, $attr, $value);

        if ($r === -1) {
            echo "Erreur : ".ldap_error($ds);
        } elseif ($r === TRUE) {
            echo "Mot de passe correct.";
        } elseif ($r === FALSE) {
            echo "Mal choisi! Mot de passe incorrect!";
        }

    } else {
        echo "Impossible de se connecter au serveur LDAP.";
    }

    ldap_close($ds);
} else {
    echo "Impossible de se connecter au serveur LDAP.";
}

```

```
}
?>
```

Attention

ldap_compare ne peut PAS être utilisé pour comparer des valeurs binaires!

Note

ldap_compare a été ajoutée en 4.0.2.

8.60.14 ldap_connect() : Se connecte à un serveur LDAP

resource **ldap_connect** (*string hostname* , *int port*)

ldap_connect retourne un identifiant positif de serveur LDAP en cas de succès, ou bien FALSE en cas d'erreur. Lorsque OpenLDAP 2.x.x est utilisé, ldap_connect retournera toujours une ressource car il ne se connectera pas mais initialisera uniquement les paramètres de connexion.

Actuellement, la connexion intervient avec le prochain appel aux fonctions ldap_* , habituellement avec la fonction ldap_bind .

ldap_connect établit une connexion avec un serveur LDAP situé sur l'hôte *hostname* et *port* . Les deux arguments sont optionnels. Sans argument, l'identifiant de la dernière connexion ouverte sera retourné. Si seul *hostname* est spécifié, le port par défaut est 389.

Si vous utilisez OpenLDAP 2.x.x, vous pouvez spécifier une URL au lieu d'un nom d'hôte. Pour utiliser LDAP avec SSL, compilez OpenLDAP 2.x.x avec le support SSL, configurez PHP avec SSL, et utilisez `ldaps://hostname/` comme nom d'hôte. Le paramètre de port *port* n'est pas utile lorsqu'utilisé avec des URL.

Note

Le support des URL et SSL a été ajouté en PHP 4.0.4.

Exemple de connexion à un serveur LDAP

```
<?php
// LDAP variables
$ldaphost = "ldap.example.com"; // votre serveur LDAP
$ldapport = 389;                // votre port de serveur LDAP

// Connecting to LDAP
$ldapconn = ldap_connect( $ldaphost, $ldapport )
            or die( "Impossible de se connecter au serveur LDAP { $ldaphost } " );

?>
```

Exemple de connexion à un serveur LDAP SSL

```
<?php
// Assurez-vous que l'hôte est correct
// et que vous avez un certificat valide
$ldaphost = "ldaps://ldap.example.com/";

// Connexion à LDAP
$ldapconn = ldap_connect( $ldaphost )
            or die( "Impossible de se connecter au serveur LDAP { $ldaphost } " );

?>
```

Voir aussi ldap_bind .

8.60.15 `ldap_count_entries()` : Compte le nombre d'entrées après une recherche

`int ldap_count_entries (resource link_identifier , resource result_identifier)`

`ldap_count_entries` retourne le nombre d'entrées trouvées dans le résultat `result_identifier` , sur la connexion `link_identifier` , ou `FALSE` en cas d'erreur.

`ldap_count_entries` retourne le nombre d'entrées stockées dans le résultat de la recherche précédente. `result_identifier` représente un résultat interne LDAP.

8.60.16 `ldap_delete()` : Efface une entrée dans un dossier

`bool ldap_delete (resource link_identifier , string dn)`

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

`ldap_delete` efface l'entrée identifiée par son `dn` sur le serveur identifié par `link_identifier` .

8.60.17 `ldap_dn2ufn()` : Convertit un DN en format UFN (User Friendly Naming)

`string ldap_dn2ufn (string dn)`

`ldap_dn2ufn` convertit le DN `dn` dans un format plus lisible humainement, en supprimant les types des noms.

8.60.18 `ldap_err2str()` : Convertit un numéro d'erreur LDAP en message d'erreur

`string ldap_err2str (int errno)`

`ldap_err2str` retourne un message d'erreur.

`ldap_err2str` retourne le message lisible expliquant l'erreur dont le numéro est `errno` . Bien que les numéros d'erreur LDAP soient standardisés, différentes bibliothèques retournent des messages différents ou même des textes d'erreur localisés. N'utilisez jamais les messages d'erreur pour identifier une erreur, mais bien les numéros.

<p>Enumerer tous les messages d'erreur LDAP</p> <pre><?php for(\$i=0; \$i<100; \$i++) { printf("Erreur numéro \$i : %s
\n", ldap_err2str(\$i)); } ?></pre>
--

Voir aussi `ldap_errno` et `ldap_error` .

8.60.19 ldap_errno() : Retourne le numéro d'erreur LDAP de la dernière commande exécutée

int **ldap_errno** (resource link_identifieur)

ldap_errno retourne le numéro d'erreur LDAP généré par la dernière commande.

ldap_errno retourne le numéro d'erreur standard, généré par la dernière commande LDAP, pour la connexion link_identifieur . Ce numéro peut être converti en message textuel avec ldap_err2str .

A moins que vous n'abaissiez suffisamment le niveau d'erreur dans php.ini (ou php3.ini), ou que vous ne préfixiez vos commandes LDAP avec @ (at) pour supprimer les affichages, les erreurs LDAP s'afficheront aussi dans le code PHP.

Genérer et intercepter une erreur

```
<?php
// Cet exemple contient une erreur, que nous intercepterons.
$lid = ldap_connect("localhost");
$bind = ldap_bind($lid);
// erreur de syntaxe dans l'expression du filtre (errno 87),
// must be "objectclass=" to work.
$res = @ldap_search($lid, "o=Myorg, c=DE", "objectclass");
if (!$res) {
    echo "LDAP-Errno: " . ldap_errno($lid) . "<br />\n";
    echo "LDAP-Error: " . ldap_error($lid) . "<br />\n";
    die("Argh!<br />\n");
}
$info = ldap_get_entries($lid, $res);
echo $info["count"] . " matching entries.<br />\n";
?>
```

Voir aussi ldap_err2str et ldap_error .

8.60.20 ldap_error() : Retourne le message LDAP de la dernière commande LDAP

string **ldap_error** (resource link_identifieur)

ldap_error retourne un message d'erreur LDAP.

ldap_error retourne le message d'erreur lié à la connexion link_identifieur . Même si les numéros d'erreur LDAP sont standardisés, différentes bibliothèques retournent différents messages, ou parfois, des messages en langue locale. Ne vous fiez pas au message d'erreur, mais bien au numéro d'erreur.

A moins que vous n'abaissiez suffisamment le niveau d'erreur dans php.ini (ou php3.ini), ou que vous ne préfixiez vos commandes LDAP avec @ pour supprimer les affichages, les erreurs LDAP s'afficheront aussi dans le code PHP.

Voir aussi ldap_err2str et ldap_errno .

8.60.21 `ldap_explode_dn()` : Sépare les différents composants d'un DN

array `ldap_explode_dn` (string `dn` , int `with_attrib`)

`ldap_explode_dn` sert à extraire les différents composants du DN `dn` . Chaque composant s'appelle un Nom Distingué Relatif (Relative Distinguished Name ou encore RDN). `ldap_explode_dn` retourne un tableau de tous ces composants. L'argument `with_attrib` sert à préciser si les RDN sont retournés seuls, ou bien avec leurs attributs. Pour obtenir les attributs en même temps que les RDN (au format attribut=valeur), donnez à `with_attrib` la valeur de 0 et, sinon, donnez lui la valeur de 1.

8.60.22 `ldap_first_attribute()` : Retourne le premier attribut

string `ldap_first_attribute` (resource `link_identifieur` , resource `result_entry_identifieur` , int `ber_identifieur`)

`ldap_first_attribute` retourne le premier attribut de l'entrée `result_entry_identifieur` , sur le serveur `link_identifieur` , ou FALSE en cas d'erreur.

Similairement à la lecture des entrées, les attributs sont lus les uns après les autres, pour une entrée. `ldap_first_attribute` retourne le premier attribut de l'entrée représentée par `result_entry_identifieur` . Les autres attributs sont lus grâce à la fonction `ldap_next_attribute` , appelée aussi souvent que nécessaire. Le paramètre `ber_identifieur` est l'identifiant de pointeur en mémoire locale. Il est passé par référence. Le même paramètre `ber_identifieur` est passé à la fonction `ldap_next_attribute` , qui modifiera ce pointeur.

Voir aussi `ldap_get_attributes` .

8.60.23 `ldap_first_entry()` : Retourne la première entrée

resource `ldap_first_entry` (resource `link_identifieur` , resource `result_identifieur`)

`ldap_first_entry` retourne la première entrée du résultat `result_identifieur` , pour la connexion `link_identifieur` , ou FALSE en cas d'erreur.

Les entrées d'un résultat LDAP sont lues séquentiellement avec les fonctions `ldap_first_entry` et `ldap_next_entry` . `ldap_first_entry` retourne la première entrée. Celles qui suivent seront lues grâce à la fonction `ldap_next_entry` , en l'appelant aussi souvent que nécessaire.

Voir aussi `ldap_get_entries` .

8.60.24 `ldap_first_reference()` : Retourne la première référence

resource `ldap_first_reference` (resource `link` , resource `result`)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.60.25 ldap_free_result() : Libère la mémoire du résultat

bool **ldap_free_result** (resource result_identifier)

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

ldap_free_result libère toute la mémoire allouée en interne pour stocker le résultat result_identifier . Si l'appel de cette fonction est omis, toute la mémoire sera libérée automatiquement à la fin du script.

Typiquement, toute la mémoire allouée pour le résultat LDAP est libérée à la fin du script. Si le script effectue des recherches intensives, qui retournent des résultats de grandes taille, ldap_free_result peut être utilisée pour réduire la consommation de mémoire.

8.60.26 ldap_get_attributes() : Lit les attributs d'une entrée

array **ldap_get_attributes** (resource link_identifier , resource result_entry_identifier)

ldap_get_attributes retourne le détail des informations d'une entrée sous la forme d'un tableau multidimensionnel ou FALSE en cas d'erreur.

ldap_get_attributes sert à simplifier la lecture des attributs et de leur valeur pour une entrée dans un résultat de recherche. La valeur retournée est un tableau multidimensionnel, avec les noms des attributs et leurs valeurs.

Une fois que vous avez repéré une entrée dans un dossier, vous pouvez obtenir plus d'informations sur elle avec cette fonction. Elle pourrait être utilisée dans le cadre d'une application qui cartographie les dossiers et les entrées. Dans de nombreuses applications, vous recherchez des entrées ayant un attribut précis, sans vous soucier des autres attributs.

Structure du tableau retourné par ldap_get_attributes

```
return_value["count"] = nombre d'attributs dans l'entrée
return_value[0] = premier attribut
return_value[n] = n-ième attribut

return_value["attribute"]["count"] = nombre de valeurs de l'attribut
return_value["attribute"][0] = première valeur de l'attribut
return_value["attribute"][i] = (i+1)-ème valeur de l'attribut
```

Affiche la liste des attributs d'une entrée

```
<?php
// $ds est une ressource de connexion valide

// $sr est une recherche valide, issue d'une opération
// précédente

$entry = ldap_first_entry($ds, $sr);

$attrs = ldap_get_attributes($ds, $entry);

echo $attrs["count"] . " attributs dans cette entrée :<p>";

for ($i=0; $i<$attrs["count"]; $i++) {
    echo $attrs[$i]."<br />";
}
?>
```

Voir aussi [ldap_first_attribute](#) et [ldap_next_attribute](#) .

8.60.27 ldap_get_dn() : Lit le DN d'une entrée

string **ldap_get_dn** (resource link_identifieur , resource result_entry_identifieur)

[ldap_get_dn](#) lit le DN de l'entrée courante dans le résultat de recherche LDAP result_entry_identifieur , ou FALSE , en cas d'erreur.

[ldap_get_dn](#) sert à lire le DN dans un résultat de recherche.

8.60.28 ldap_get_entries() : Lit toutes les entrées du résultat

array **ldap_get_entries** (resource link_identifieur , resource result_identifieur)

[ldap_get_entries](#) retourne toutes les entrées du résultat result_identifieur , sous la forme d'un tableau multi-dimensionnel, ou FALSE en cas d'erreur.

[ldap_get_entries](#) sert à simplifier la lecture des entrées dans un résultat, ainsi que celle des attributs correspondants. La structure du tableau retourné est la suivante :

L'index d'attribut est converti en minuscules (les attributs sont sensibles à la casse pour les serveurs de dossiers, mais ne le sont pas lorsqu'ils sont utilisés comme index de tableaux).

Structure du tableau retourné par ldap_get_entries
return_value["count"] = nombre d'entrées dans le résultat
return_value[0] : se réfère aux détails de la première entrée
return_value[n]["dn"] = DN de la n-ième entrée du résultat
return_value[n]["count"] = nombre d'attributs de la n-ième entrée
return_value[n][m] = m-ième attribut de la i-ème entrée
return_value[n]["attribute"]["count"] = nombre de valeur des attributs de la i-ième entrée
return_value[n]["attribute"][m] = m-ième valeur de la n-ième entrée

Voir aussi [ldap_first_entry](#) et [ldap_next_entry](#) .

8.60.29 ldap_get_option() : Lit/écrit la valeur courante d'une option

bool **ldap_get_option** (resource link_identifieur , int option , mixed retval)

[ldap_get_option](#) modifie la valeur de l'option option à la valeur retval pour la connexion link_identifieur . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre option peut prendre l'une des valeurs suivantes : LDAP_OPT_DEREF, LDAP_OPT_SIZELIMIT, LDAP_OPT_TIMELIMIT, LDAP_OPT_PROTOCOL_VERSION, LDAP_OPT_ERROR_NUMBER, LDAP_OPT_REFERRALS, LDAP_OPT_RESTART, LDAP_OPT_HOST_NAME, LDAP_OPT_ERROR_STRING, LDAP_OPT_MATCHED_DN. Elles sont décrites dans [draft-ietf-ldapext-ldap-c-api-xx.txt](#)

Note

ldap_get_option n'est disponible qu'avec OpenLDAP 2.x.x OU Netscape Directory SDK x.x, et a été ajoutée en PHP 4.0.4.

Vérification de la version du protocole avec ldap_get_option

```
<?php
// $ds doit être une ressource de connexion valide
if (ldap_get_option($ds, LDAP_OPT_PROTOCOL_VERSION, $version))
    echo "Nous utilisons le protocole version $version";
else
    echo 'Impossible de déterminer la version du protocole.';
?>
```

Voir aussi ldap_set_option .

8.60.30 ldap_get_values_len() : Lit toutes les valeurs binaires d'une entrée

array **ldap_get_values_len** (resource link_identifieur , resource result_entry_identifieur , string attribute)

ldap_get_values_len retourne un tableau de valeurs pour l'attribut attribute en cas de succès, et FALSE en cas d'erreur.

ldap_get_values_len sert à lire toutes les valeurs de l'attribut attribute dans le résultat result_entry_identifieur . L'entrée utilisée est l'entrée courante du résultat de recherche LDAP result_entry_identifieur . Le nombre de valeurs retournées est disponible à l'index 'count' du tableau retourné. Les valeurs sont accessibles individuellement, avec les index numériques du tableau. L'indexation commence à 0.

Cette fonction s'utilise exactement comme ldap_get_values , hormis le fait qu'elle gère les données binaires, et non pas les chaînes de caractères.

Note

Cette fonction a été ajoutée en PHP 4.0.

8.60.31 ldap_get_values() : Lit toutes les valeurs d'une entrée LDAP

array **ldap_get_values** (resource link_identifieur , resource result_entry_identifieur , string attribute)

ldap_get_values retourne un tableau de valeurs pour l'attribut attribute , ou FALSE en cas d'erreur.

ldap_get_values sert à lire toutes les valeurs de l'attribut attribute , dans l'entrée courante du résultat de recherche result_entry_identifieur , pour la connexion link_identifieur . Le nombre de valeurs retournées est disponible à l'index 'count' du tableau retourné. Les valeurs sont accessibles individuellement, avec les index numériques du tableau. L'indexation commence à 0.

L'utilisation de la fonction result_entry_identifieur requiert un résultat de recherche, et doit donc être précédée d'une recherche LDAP, et de l'une des fonctions permettant d'accéder à une entrée.

Votre application doit contenir des informations permettant de lire certains attributs (comme "nom" ou "mail"), ou bien vous devrez utiliser la fonction ldap_get_attributes pour savoir quels sont les attributs qui existent pour une entrée donnée.

LDAP permet plus d'une entrée par attribut, ce qui permet de stocker plusieurs adresse emails par personne, tout en n'utilisant qu'une étiquette "mail" :

Structure du tableau retourné par <code>ldap_get_values</code>
<pre>return_value["count"] = nombre de valeurs de l'attribut return_value[0] = première valeur de l'attribut return_value[i] = n-ième valeur de l'attribut</pre>
Liste toutes les valeurs de l'attribut "mail" d'une entrée
<pre><?php // \$ds doit être une ressource de connexion valide // \$sr doit être une ressource de résultat valide, obtenue avec une des fonctions de // recherche LDAP. // \$entry est une entrée LDAP valide, obtenue avec une des fonctions // LDAP qui retourne une entrée \$values = ldap_get_values(\$ds, \$entry,"mail"); echo \$values["count"] . " adresses email pour cette entrée.
"; for (\$i=0; \$i < \$values["count"]; \$i++) { echo \$values[\$i] . "
"; } ?></pre>

8.60.32 `ldap_list()` : Recherche dans un niveau

resource **ldap_list** (resource link_identifier , string base_dn , string filter , array attributes , int attrsonly , int sizelimit , int timelimit , int deref)

`ldap_list` retourne un identifiant de résultat ou bien FALSE en cas d'erreur.

`ldap_list` effectue une recherche avec le filtre filter dans le dossier base_dn avec l'option LDAP_SCOPE_ONELEVEL .

LDAP_SCOPE_ONELEVEL signifie que la recherche ne peut retourner des entrées que dans le niveau qui est immédiatement sous le niveau base_dn (c'est l'équivalent de la commande "ls", pour obtenir la liste des fichiers et dossiers du dossier courant).

`ldap_list` prend 5 paramètres optionnels. Voir `ldap_search` pour connaître le détail de leur utilisation.

Note
Ces paramètres optionnels ont été ajoutés en PHP 4.0.2 : attrsonly , sizelimit , timelimit et deref .
Produit une liste de tous les services d'une société avec <code>ldap_list</code>
<pre><?php // \$ds doit être une ressource de connexion valide \$basedn = "o=Ma Compagnie, c=FR"; \$justthese = array("ou"); \$sr=ldap_list(\$ds, \$basedn, "ou=*", \$justthese); \$info = ldap_get_entries(\$ds, \$sr); for (\$i=0; \$i<\$info["count"]; \$i++){ echo \$info[\$i]["ou"][0] ; } }</pre>

?>

Note

Depuis PHP 4.0.5, il est aussi possible de faire des recherches parallèles. Reportez-vous à ldap_search pour plus de détails.

8.60.33 ldap_mod_add() : Ajoute un attribut à l'entrée courante

bool **ldap_mod_add** (resource link_identifieur , string dn , array entry)

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

[ldap_mod_add](#) ajoute l'attribut entry à l'entrée dn . Elle effectue la modification au niveau attribut, par opposition au niveau objet. Les additions au niveau objet sont réalisées par [ldap_add](#) .

Note

Cette fonction gère les chaînes binaires.

8.60.34 ldap_mod_del() : Efface un attribut à l'entrée courante

bool **ldap_mod_del** (resource link_identifieur , string dn , array entry)

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

[ldap_mod_del](#) efface l'attribut entry de l'entrée dn . Elle effectue la modification au niveau attribut, par opposition au niveau objet. Les additions au niveau objet sont réalisées par [ldap_delete](#) .

8.60.35 ldap_mod_replace() : Remplace un attribut dans l'entrée courante

bool **ldap_mod_replace** (resource link_identifieur , string dn , array entry)

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

[ldap_mod_replace](#) remplace l'attribut entry de l'entrée dn . Elle effectue le remplacement au niveau attribut, par opposition au niveau objet. Les additions au niveau objet sont réalisées par [ldap_modify](#) .

Note

Cette fonction gère les chaînes binaires.

8.60.36 ldap_modify() : Modifie une entrée LDAP

bool **ldap_modify** (resource link_identifieur , string dn , array entry)

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

[ldap_modify](#) modifie l'entrée identifiée par dn , avec les valeurs fournies dans entry . La structure de entry est la même que détaillée dans [ldap_add](#) .

Note

Cette fonction gère les chaînes binaires.

8.60.37 ldap_next_attribute() : Lit l'attribut suivant

string **ldap_next_attribute** (resource link_identifieur , resource result_entry_identifieur , resource ber_identifieur)

ldap_next_attribute retourne l'attribut suivant en cas de succès et, sinon, une erreur.

ldap_next_attribute sert à lire tous les attributs d'une entrée. Le pointeur interne est géré par ber_identifieur . Il est passé par référence à la fonction. Le premier appel à ldap_next_attribute est fait avec le result_entry_identifieur retourné par ldap_first_attribute .

Voir aussi ldap_get_attributes .

8.60.38 ldap_next_entry() : Lit la prochaine entrée

resource **ldap_next_entry** (resource link_identifieur , resource result_entry_identifieur)

ldap_next_entry retourne un identifiant d'entrée, pour la prochaine entrée dans le résultat result_entry_identifieur . La première entrée a été lue avec la fonction ldap_first_entry . S'il n'y a plus d'entrées dans le résultat, ldap_next_entry retourne FALSE .

ldap_next_entry sert à lire successivement les entrées stockées dans le résultat result_entry_identifieur . Des appels répétés à ldap_next_entry retourneront toutes les entrées jusqu'à ce qu'il n'y en ait plus. Le premier appel doit être fait avec la fonction ldap_first_entry . Le paramètre result_entry_identifieur est celui qui a été retourné par la fonction ldap_first_entry .

Voir aussi ldap_get_entries .

8.60.39 ldap_next_reference() : Lit la référence suivante

resource **ldap_next_reference** (resource link , resource entry)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.60.40 ldap_parse_reference() : Extrait les informations d'une référence d'entrée

bool **ldap_parse_reference** (resource link , resource entry , array referrals)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.60.41 ldap_parse_result() : Extrait des informations d'un résultat

bool **ldap_parse_result** (resource link , resource result , int errcode , string matcheddn , string errmsg , array referrals)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.60.42 `ldap_read()` : Lit une entrée

resource **ldap_read** (resource link_identifier , string base_dn , string filter , array attributes , int attrsonly , int sizelimit , int timelimit , int deref)

`ldap_read` retourne un identifiant de ressource, ou FALSE en cas d'erreur.

`ldap_read` effectue une recherche avec le filtre filter dans le dossier base_dn avec la configuration LDAP_SCOPE_BASE . C'est équivalent à lire une entrée dans un dossier.

Un filtre ne peut être vide. Si vous voulez lire toutes les informations d'une entrée, utilisez le filtre " objectClass=* ". Si vous savez quels sont les types qui sont utilisés dans le serveur de dossiers, vous pouvez aussi utiliser un filtre approprié, comme " objectClass=inetOrgPerson ".

`ldap_read` prend 5 paramètres optionnels. Reportez-vous à `ldap_search` pour plus de détails.

Note

Ces paramètres optionnels ont été ajoutés en PHP 4.0.2 : attrsonly , sizelimit , timelimit et deref . Depuis PHP 4.0.5, il est aussi possible d'effectuer des recherches parallèles. Voyez <code>ldap_search</code> pour plus de détails.

8.60.43 `ldap_rename()` : Modifie le nom d'une entrée

bool **ldap_rename** (resource link_identifier , string dn , string newrdn , string newparent , bool deleteoldrdn)

`ldap_rename` modifie l'entrée dn , autant pour son nom que pour sa localisation. Le nouveau RDN est spécifié avec newrdn et le nouveau père est spécifié avec newparent . Si le paramètre deleteoldrdn vaut TRUE , l'ancienne valeur RDN est supprimée. Sinon elle est conservée comme une valeur non distinguée. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

<code>ldap_rename</code> ne fonctionne actuellement qu'avec LDAPv3. Vous pouvez être obligé d'utiliser <code>ldap_set_option</code> avant de vous lier pour pouvoir utiliser LDAPv3. Cette fonction est uniquement disponible lorsque vous utilisez OpenLDAP 2.x.x OU Netscape Directory SDK x.x. Elle a été ajoutée en PHP 4.0.5.
--

8.60.44 `ldap_sasl_bind()` : Authentification au serveur LDAP en utilisant SASL

bool **ldap_sasl_bind** (resource link , string binddn , string password , string sasl_mech , string sasl_realm , string sasl_authz_id , string props)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Note

Conditions d'utilisation

`ldap_sasl_bind` nécessite le support SASL (`sasl.h`). Assurez-vous que l'option de configuration `--with-ldap-sasl` est utilisée lors de la compilation de PHP, sinon, cette fonction ne sera pas définie.

8.60.45 `ldap_search()` : Recherche sur le serveur LDAP

resource **ldap_search** (resource link_identifier , string base_dn , string filter , array attributes , int attrsonly , int sizelimit , int timelimit , int deref)

ldap_search retourne un identifiant de résultat, ou bien FALSE en cas d'erreur.

`ldap_search` effectue une recherche avec le filtre filter dans le dossier base_dn avec le paramétrage LDAP_SCOPE_SUBTREE . C'est l'équivalent d'une recherche dans le dossier. base_dn spécifie le DN de base du dossier.

Il y a un quatrième paramètre optionnel, qui peut être fourni pour restreindre les attributs et valeurs retournées par le serveur, afin de les adapter à vos besoins. C'est bien plus efficace que le comportement par défaut (qui retourne tous les attributs et leurs valeurs associées). L'utilisation de ce quatrième paramètre est donc vivement recommandée.

Le quatrième paramètre est un tableau de chaînes PHP avec les attributs demandés, par exemple : `array("mail","sn","cn")` . Notez que le "dn" est toujours retourné, indépendamment des attributs demandés.

Notez aussi que certains serveurs de dossiers sont configurés pour retourner un nombre limité d'entrées. Si cette situation survient, le serveur indiquera qu'il a renvoyé un résultat partiel. C'est aussi le cas si le sixième paramètre sizelimit a été utilisé pour limiter le nombre d'entrées retournées.

Le cinquième paramètre attrsonly doit être mis à 1 si seuls les types d'attributs sont désirés. Si ce paramètre est mis à 0, les types d'attributs et leurs valeurs seront retournées : c'est le comportement par défaut.

Avec le sixième paramètre sizelimit , il est possible de limiter le nombre d'entrées lues dans le résultat. En utilisant 0, on indique qu'il n'y a pas de limite. NOTE : ce paramètre NE peut PAS remplacer la configuration du serveur. Vous pouvez au mieux baisser encore cette limite.

Le septième paramètre timelimit permet de spécifier le nombre de secondes de durée de la recherche. En utilisant la valeur 0, la recherche pourra durer indéfiniment. NOTE : ce paramètre NE peut PAS remplacer la configuration du serveur. Vous pouvez au mieux baisser encore cette limite.

Le huitième paramètre deref spécifie comment les alias doivent être gérés durant la recherche. Ce paramètre peut prendre l'une des valeurs suivantes :

- LDAP_DEREF_NEVER - (par défaut) les alias ne sont jamais "déréférencés".
- LDAP_DEREF_SEARCHING - les alias sont "déréférencés" durant la recherche, mais pas durant la localisation du dossier de recherche.
- LDAP_DEREF_FINDING - les alias sont "déréférencés" durant la localisation du dossier de recherche, mais pas durant la recherche.
- LDAP_DEREF_ALWAYS - les alias sont toujours "déréférencés".

Note

Ces paramètres optionnels ont été ajoutés en 4.0.2 : attrsonly , sizelimit , timelimit et deref .

Le filtre de recherche peut être simple ou avancé, et utiliser ces opérateurs booléen au format décrit dans la documentation LDAP (reportez-vous à [Netscape Directory SDK](#) pour plus d'informations sur

les filtres).

L'exemple ci-dessous lit le nom du service, le nom, le prénom et l'email des employés de la société "Ma Compagnie", dont le nom ou le prénom contient la sous-chaîne : \$person. Cet exemple illustre l'utilisation de filtres pour indiquer au serveur de faire une recherche sur deux attributs.

```

Recherche LDAP
<?php
// $ds doit être une ressource de connexion valide

// $person est un nom ou une partie de nom (par exemple, "Jean")

$dn = "o=Ma Compagnie, c=FR";
$filter="(|(sn=$person*)(givenname=$person*))";
$justthese = array( "ou", "sn", "givenname", "mail");

$sr=ldap_search($ds, $dn, $filter, $justthese);

$info = ldap_get_entries($ds, $sr);

echo $info["count"]." entrées trouvées.\n";
?>

```

Depuis PHP 4.0.5, il est possible de faire des recherches parallèles. Pour cela, vous devez utiliser un tableau de ressources de connexion comme premier argument, au lieu d'une ressource simple. Si vous ne voulez pas utiliser le même dossier de base DN et le même filtre pour toutes les recherches, vous pouvez aussi utiliser un tableau de DN et de filtres. Ces tableaux doivent alors être de la même taille que le tableau de ressources de connexion : la première entrée du tableau de ressources sera utilisée avec le premier DN et le premier filtre, etc. Lorsque vous faites une recherche parallèle, un tableau de résultats est retourné, sauf en cas d'erreur, auquel cas l'entrée contiendra FALSE . C'est très proche du comportement habituel, hormis le fait qu'un résultat est toujours retourné lorsqu'une recherche est faite. Il y a de rares cas où une recherche renvoi FALSE , alors qu'une autre en parallèle retourne un identifiant.

8.60.46 ldap_set_option() : Modifie la valeur d'une option LDAP

bool **ldap_set_option** (resource link_identifier , int option , mixed newval)

ldap_set_option modifie la valeur de l'option option en remplaçant la valeur courante par newval . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

Le paramètre option peut prendre l'une des valeurs suivantes : LDAP_OPT_DEREF, LDAP_OPT_SIZELIMIT, LDAP_OPT_TIMELIMIT, LDAP_OPT_PROTOCOL_VERSION, LDAP_OPT_ERROR_NUMBER, LDAP_OPT_REFERRALS, LDAP_OPT_RESTART, LDAP_OPT_HOST_NAME, LDAP_OPT_ERROR_STRING, LDAP_OPT_MATCHED_DN, LDAP_OPT_SERVER_CONTROLS, LDAP_OPT_CLIENT_CONTROLS. Voici une brève description des valeurs ci-dessus, dans le fichier [draft-ietf-ldapext-ldap-c-api-xx.txt](#) .

Les options LDAP_OPT_DEREF, LDAP_OPT_SIZELIMIT, LDAP_OPT_TIMELIMIT, LDAP_OPT_PROTOCOL_VERSION et LDAP_OPT_ERROR_NUMBER doivent être des entiers, LDAP_OPT_REFERRALS et LDAP_OPT_RESTART doivent être des booléens, et LDAP_OPT_HOST_NAME, LDAP_OPT_ERROR_STRING et LDAP_OPT_MATCHED_DN doivent être des chaînes. Le premier exemple illustre leur utilisation. Les options LDAP_OPT_SERVER_CONTROLS et LDAP_OPT_CLIENT_CONTROLS requièrent une liste de contrôles, ce qui signifie que la valeur doit être un tableau de contrôles. Un contrôle est constitué d'un **oid** l'identifiant, une valeur optionnelle **value** , et un flag optionnel de "criticalité" (**criticality**).

En PHP, un contrôle est défini sous la forme d'un tableau, donc les clés sont **oid** avec une chaîne comme valeur, et deux clés optionnelles. Ces clés sont **value** avec une chaîne comme valeur, et **iscritical** avec une valeur booléenne. Par défaut, **iscritical** vaut FALSE . Reportez-vous au second exemple pour une illustration.

Note

Cette fonction n'est disponible que lorsque vous utilisez OpenLDAP 2.x.x ou Netscape Directory SDK x.x. Elle a été ajoutée en PHP 4.0.4.
--

Modification de la version du protocole

```
<?php
// $ds doit être une ressource de connexion valide
if (ldap_set_option($ds, LDAP_OPT_PROTOCOL_VERSION, 3)) {
    echo 'Version LDAPv3';
} else {
    echo 'Impossible de modifier la version du protocole à 3';
}
?>
```

Modification des contrôles du serveur

```
<?php
// $ds doit être une ressource de connexion valide
// controle sans valeur
$ctrl1 = array("oid" => "1.2.752.58.10.1", "iscritical" => TRUE);
// iscritical vaut par défaut FALSE
$ctrl2 = array("oid" => "1.2.752.58.1.10", "value" => "magic");
// essayez d'utiliser les deux contrôles
if (!ldap_set_option($ds, LDAP_OPT_SERVER_CONTROLS, array($ctrl1, $ctrl2))) {
    echo 'Impossible de modifier les contrôles du serveur';
}
?>
```

Voir aussi [ldap_get_option](#) .

8.60.47 ldap_set_rebind_proc() : Configure une fonction de callback pour refaire des liaisons lors de recherche de référants

bool **ldap_set_rebind_proc** (resource link , callback callback)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.60.48 ldap_sort() : Trie les entrées d'un résultat LDAP

bool **ldap_sort** (resource link , resource result , string sortfilter)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.60.49 ldap_start_tls() : Démarre TLS

bool **ldap_start_tls** (resource link)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.60.50 ldap_t61_to_8859() : Convertit les caractères t6 en caractères 8859

string **ldap_t61_to_8859** (string value)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.60.51 ldap_unbind() : Déconnecte d'un serveur LDAP

bool **ldap_unbind** (resource link_identifier)

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

ldap_unbind assure la déconnexion du serveur LDAP.

8.61 libxml

8.61.1 Introduction

Ces fonctions et constantes sont disponibles depuis PHP 5.1.0 et si vous avez compilé PHP avec les extensions basées sur libxml, c'est à dire [DOM](#) , [SimpleXML](#) et [XSLT](#) .

8.61.2 Pré-requis

Cette extension requiert [libxml](#) >= 2.6.0.

8.61.3 Classes pré-définies

8.61.3.1 LibXML_Error

8.61.3.1.1 Propriétés

- code - le code erreur
- column - la colonne où l'erreur est survenue. Notez que cette propriété n'est pas entièrement implémentée dans libxml et donc, 0 est toujours retourné.
- file - le nom du fichier, ou vide si le XML a été chargé depuis une chaîne.
- level - le degré de l'erreur (une des constantes suivantes : LIBXML_ERR_WARNING , LIBXML_ERR_ERROR ou LIBXML_ERR_FATAL)
- line - la ligne où l'erreur est survenue.
- message - le message de l'erreur

8.61.4 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

LIBXML_COMPACT (integer)

Active l'optimisation de l'allocation de petits noeuds. Ceci pourrait augmenter la rapidité de votre application sans avoir besoin de changer votre code.

Note

Seulement disponible dans Libxml >= 2.6.21
--

LIBXML_DTDATTR (integer)

Attribut de DTD par défaut

LIBXML_DTDLOAD (integer)

Charge le sous-ensemble externe

LIBXML_DTDVALID (integer)

Valide avec la DTD

LIBXML_NOBLANKS (integer)

Suppression des noeuds vides

LIBXML_NOCDATA (integer)

Fusion des CDATA en noeuds de texte

LIBXML_NOEMPTYTAG (integer)

Agrandi les tags vides (par exemple,
 en
</br>)

Note
Cette option est présentement seulement disponible dans les fonctions <code>DOMDocument->save()</code> et <code>DOMDocument->saveXML()</code> .

LIBXML_NOENT (integer)

Substitution des entités

LIBXML_NOERROR (integer)

Suppression du rapport d'erreur

LIBXML_NONET (integer)

Désactivation du réseau lors du chargement de document

LIBXML_NOWARNING (integer)

Suppression des rapports d'alerte

LIBXML_NOXMLDECL (integer)

Annule la déclaration XML lors de la sauvegarde du document

Note
Seulement disponible dans Libxml >= 2.6.21

LIBXML_NSCLEAN (integer)

Suppression des espaces de noms redondants

LIBXML_XINCLUDE (integer)

Implementation de la substitution XInclude

LIBXML_ERR_ERROR (integer)

Erreur non-fatale

LIBXML_ERR_FATAL (integer)

Erreur fatale

LIBXML_ERR_NONE (integer)

Aucune erreur

LIBXML_ERR_WARNING (integer)

Une alerte simple

LIBXML_VERSION (integer)

libxml version sous la forme 20605 ou 20617

LIBXML_DOTTED_VERSION (string)

libxml version sous la forme 2.6.5 ou 2.6.17

- [Introduction](#)
- [Pré-requis](#)
- [Classes pré-définies](#)
- [Constantes pré-définies](#)
- [libxml_clear_errors](#)
- [libxml_get_errors](#)
- [libxml_get_last_error](#)
- [libxml_set_streams_context](#)
- [libxml_use_internal_errors](#)

8.61.6 libxml_get_errors() : Lit le tableau d'erreurs

array `libxml_get_errors` (void)

Retrieve array of errors.

8.61.7 libxml_get_last_error() : Lit la dernière erreur libxml

LibXMLError **libxml_get_last_error** (void)

libxml_get_last_error lit la dernière erreur libxml.

8.61.8 libxml_set_streams_context() : Configure le contexte de flux pour la prochaine opération libxml

void **libxml_set_streams_context** (resource streams_context)

libxml_set_streams_context configure le contexte de flux pour la prochaine opération libxml.

8.61.9 libxml_use_internal_errors() : Désactive le rapport d'erreur libxml et les stocke pour lecture ultérieure

bool **libxml_use_internal_errors** (*bool use_errors*)

libxml_use_internal_errors vous permet de désactiver le gestionnaire d'erreurs libxml standard, et d'activer votre propre gestionnaire d'erreur.

8.62 Fonctions LZF

8.62.1 Introduction

LZF est un algorithme de compression très rapide, idéal pour économiser de l'espace disque sans trop de perte de temps. Il peut être optimisé pour la vitesse ou pour la compression au moment de la compilation.

8.62.2 Installation

Cette extension PECL n'est pas intégrée à PHP. Des informations comme les notes sur les nouvelles versions, les téléchargements, les sources des fichiers, les informations concernant les mainteneurs ainsi qu'un CHANGELOG, peuvent être trouvées ici : <http://pecl.php.net/package/lzf> .

Si vous voulez utiliser ces fonctions, vous devez compiler PHP avec le support lzf en utilisant l'option de configuration `--with-lzf[=DIR]` . Vous pouvez également passer l'option `--enable-lzf-better-compression` pour optimiser LZF d'un point de vue d'espace au détriment de la vitesse.

Les utilisateurs de Windows doivent activer la bibliothèque `php_lzf.dll` dans le `php.ini` pour pouvoir utiliser ces fonctions. Vous pouvez télécharger ces bibliothèques DLL pour ces extensions PECL depuis la page de téléchargement de PHP sous sur la page <http://snaps.php.net/> .

- [Introduction](#)
- [Installation](#)
- [lzf_compress](#)
- [lzf_decompress](#)
- [lzf_optimized_for](#)

8.62.4 `lzf_decompress()` : Décompression LZF

string `lzf_decompress` (string data)

`lzf_compress` décompresse les données contenues dans le paramètre `arg` .

8.62.5 `lzf_optimized_for()` : Détermine le mode d'optimisation de l'extension LZF

int `lzf_optimized_for` (void)

`lzf_optimized_for` détermine le mode d'optimisation de l'extension LZF.

8.63 Mail

8.63.1 Introduction

La fonction `mail` permet d'envoyer un mail.

8.63.2 Pré-requis

Pour que la fonction `mail` soit disponible, il faut que PHP ait accès au service sendmail sur le serveur, au moment de la compilation. Si vous utilisez un autre programme de mail, comme qmail ou postfix, assurez-vous d'utiliser les bonnes API. PHP va commencer à chercher sendmail dans votre PATH, puis, dans les dossiers suivants : `/usr/bin:/usr/sbin:/usr/etc:/etc:/usr/ucblib:/usr/lib`. Il est hautement recommandé d'avoir sendmail de disponible dans votre PATH. De plus, l'utilisateur qui compile PHP doit avoir le droit d'accéder à l'exécutable sendmail.

8.63.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.63.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable	Historique
SMTP	"localhost"	PHP_INI_ALL	
smtp_port	"25"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
sendmail_from	NULL	PHP_INI_ALL	
sendmail_path	NULL	PHP_INI_SYSTEM	

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

`smtp` string

Sous Windows seulement : adresse DNS ou adresse IP du SMTP que PHP doit utiliser pour envoyer un mail avec la fonction `mail`.

`smtp_port` int

Sous Windows seulement : numéro de port à utiliser pour se connecter au serveur SMTP lors de l'envoi de mail avec la fonction `mail` ; par défaut, c'est 25. Uniquement disponible depuis PHP 4.3.0.

`sendmail_from` string

Sous Windows seulement : valeur du champ "From:" qui doit être utilisée lors de l'envoi de mail.

`sendmail_path` string

Localisation du programme sendmail : habituellement `/usr/sbin/sendmail` ou `/usr/lib/sendmail`. configure essaye de repérer la présence de sendmail par lui-même, et affecte ce résultat par défaut. En cas de problème de localisation, vous pouvez établir une nouvelle valeur par défaut ici.

Tout système n'utilisant pas sendmail doit établir cette directive à la valeur chemin du programme de substitution qui remplace le serveur de mail, si celui-ci existe, par exemple, Qmail . Dans ce cas-là, vous devez mettre : `/var/qmail/bin/sendmail` .

qmail-inject ne requiert aucune option pour traiter correctement le mail.

Cette directive fonctionne également sous Windows. Si elle est définie, `smtp` , `smtp_port` et `sendmail_from` sont ignorés et la commande spécifiée est exécutée.

8.63.5 Types de ressources

Cette extension ne définit aucune ressource.

8.63.6 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [ezmlm_hash](#)
- [mail](#)

8.63.8 mail() : Envoi un mail

bool **mail** (string *to* , string *subject* , string *message* , string *additional_headers* , string *additional_parameters*)

Envoi un mail.

8.64 Traitement d'email

8.64.1 Introduction

Attention

Ce module est **EXPERIMENTAL** . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, **SANS PREAVIS!** Soyez-en conscient, et utilisez ce module à vos risques et périls.

Cette extension a été déplacée dans le module PECL et ne sera plus intégrée dans PHP à partir de PHP 4.2.0.

8.64.2 Installation

Cette extension PECL n'est pas intégrée à PHP. Des informations comme les notes sur les nouvelles versions, les téléchargements, les sources des fichiers, les informations concernant les mainteneurs ainsi qu'un CHANGELOG, peuvent être trouvées ici :

<http://pecl.php.net/package/mailparse> .

Si vous voulez utiliser ces fonctions, vous devez compiler PHP avec le support mailparse en utilisant l'option de configuration `--enable-mailparse` .

Les utilisateurs de Windows doivent activer la bibliothèque `php_mailparse.dll` dans le `php.ini` pour pouvoir utiliser ces fonctions. Vous pouvez télécharger ces bibliothèques DLL pour ces extensions PECL depuis la page de téléchargement de PHP sous sur la page <http://snaps.php.net/> .

- [Introduction](#)
- [Installation](#)
- [mailparse_determine_best_xfer_encoding](#)
- [mailparse_msg_create](#)
- [mailparse_msg_extract_part_file](#)
- [mailparse_msg_extract_part](#)
- [mailparse_msg_free](#)
- [mailparse_msg_get_part_data](#)
- [mailparse_msg_get_part](#)
- [mailparse_msg_get_structure](#)
- [mailparse_msg_parse_file](#)
- [mailparse_msg_parse](#)
- [mailparse_rfc822_parse_addresses](#)
- [mailparse_stream_encode](#)
- [mailparse_uudecode_all](#)

8.64.4 `mailparse_msg_create()` : Retourne un analyseur destiné au traitement d'un message

resource `mailparse_msg_create` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, **SANS**

PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.64.5 `mailparse_msg_extract_part_file()` : Extrait et décode une section de message

string `mailparse_msg_extract_part_file` (resource `rfc2045` , string `filename` , *callback* `callbackfunc`)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.64.6 `mailparse_msg_extract_part()` : Extrait et décode une section de message

void `mailparse_msg_extract_part` (resource `rfc2045` , string `msgbody` , *callback* `callbackfunc`)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Si la fonction de callback n'est pas spécifiée, le contenu sera envoyé à la sortie standard.

8.64.7 `mailparse_msg_free()` : Détruit un analyseur créé par `mailparse_msg_create`

bool `mailparse_msg_free` (resource `rfc2045buf`)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.64.8 mailparse_msg_get_part_data() : Retourne un tableau associatif avec des informations sur le message

array mailparse_msg_get_part_data (resource rfc2045)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.64.9 mailparse_msg_get_part() : Retourne une ressource sur une section d'un message MIME

resource mailparse_msg_get_part (resource rfc2045 , string mimesection)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.64.10 mailparse_msg_get_structure() : Retourne un tableau avec les différentes sections MIME du message

array mailparse_msg_get_structure (resource rfc2045)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.64.11 mailparse_msg_parse_file() : Analyse un fichier et retourne une ressource représentant la structure du fichier

resource mailparse_msg_parse_file (string filename)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.64.12 mailparse_msg_parse() : Analyse incrémentalement des données dans un buffer

bool mailparse_msg_parse (resource rfc2045buf , string data)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.64.13 mailparse_rfc822_parse_addresses() : Analyse les adresses et retourne un tableau contenant ces données

array mailparse_rfc822_parse_addresses (string addresses)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.64.14 mailparse_stream_encode() : Lit les données dans un fichier, applique l'encodage et envoie le résultat à destfp

bool mailparse_stream_encode (resource sourcefp , resource destfp , string encoding)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.64.15 mailparse_uudecode_all() : Scanne les données du fichier et extrait tous les fichiers encodés qui s'y trouvent

array mailparse_uudecode_all (resource fp)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Retourne un tableau contenant la liste des fichiers.

8.65 Mathématiques

8.65.1 Introduction

Ces fonctions ne sont capables de manipuler que des entiers entier , ou nombres à virgule flottante (float). Si vous avez besoin de manipuler des nombres plus grands, reportez-vous aux fonctions mathématiques sur des nombres de grande taille .

Voir aussi la page du manuel sur les opérateurs arithmétiques .

8.65.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.65.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.65.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.65.5 Types de ressources

Cette extension ne définit aucune ressource.

8.65.6 Constantes pré-définies

Les constantes listées ici sont toujours disponibles dans PHP.

Constante	Valeur	Description
M_PI	3.14159265358979323846	Pi
M_E	2.7182818284590452354	e
M_LOG2E	1.4426950408889634074	log ₂ e
M_LOG10E	0.43429448190325182765	log ₁₀ e
M_LN2	0.69314718055994530942	log _e 2
M_LN10	2.30258509299404568402	log _e 10
M_PI_2	1.57079632679489661923	pi/2
M_PI_4	0.78539816339744830962	pi/4
M_1_PI	0.31830988618379067154	1/pi
M_2_PI	0.63661977236758134308	2/pi
M_SQRTPI	1.77245385090551602729	sqrt(pi) [4.0.2]
M_2_SQRTPI	1.12837916709551257390	2/sqrt(pi)

M_SQRT2	1.41421356237309504880	sqrt(2)
M_SQRT3	1.73205080756887729352	sqrt(3) [4.0.2]
M_SQRT1_2	0.70710678118654752440	1/sqrt(2)
M_LNPI	1.14472988584940017414	log_e(pi) [4.0.2]
M_EULER	0.57721566490153286061	Euler constant [4.0.2]

Seule M_PI est disponible dans les versions de PHP jusqu'à 4.0.0. Toutes les autres constantes ont été ajoutées à partir de PHP 4.0.0, et les autres ont été indiquées avec [4.0.2].

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [abs](#)
- [acos](#)
- [acosh](#)
- [asin](#)
- [asinh](#)
- [atan](#)
- [atan2](#)
- [atanh](#)
- [base_convert](#)
- [bindec](#)
- [ceil](#)
- [cos](#)
- [cosh](#)
- [decbin](#)
- [dechex](#)
- [decoct](#)
- [deg2rad](#)
- [exp](#)
- [expm1](#)
- [floor](#)
- [fmod](#)
- [getrandmax](#)
- [hexdec](#)
- [hypot](#)
- [is_finite](#)
- [is_infinite](#)
- [is_nan](#)
- [lg_value](#)
- [log](#)
- [log10](#)
- [log1p](#)
- [max](#)
- [min](#)
- [mt_getrandmax](#)
- [mt_rand](#)
- [mt_srand](#)
- [octdec](#)
- [pi](#)
- [pow](#)

- [rad2deg](#)
- [rand](#)
- [round](#)
- [sin](#)
- [sinh](#)
- [sqrt](#)
- [srand](#)
- [tan](#)
- [tanh](#)

8.65.8 acos() : arc cosinus

float **acos** (float arg)

[acos](#) retourne l'arc cosinus de arg (arg en radians). [acos](#) est la fonction inverse de [cos](#) , ce qui signifie que $a = \cos(\text{acos}(a))$ pour toute valeur qui soit dans l'intervalle de validité de [acos](#) .

Voir aussi [acosh](#) , [asin](#) et [atan](#) .

8.65.9 acosh() : Arc cosinus hyperbolique

float **acosh** (float arg)

[acosh](#) retourne l'arc cosinus hyperbolique de arg , c'est à dire la valeur dont le cosinus hyperbolique est arg .

Note
Cette fonction n'est pas implémentée sous Windows.

Voir aussi [acos](#) , [asinh](#) et [atanh](#) .

8.65.10 asin() : arc sinus

float **asin** (float arg)

[asin](#) retourne l'arc sinus de arg (arg en radians). [asin](#) est la fonction inverse de [sin](#) , ce qui signifie que $a = \sin(\text{asin}(a))$ pour toute valeur qui soit dans l'intervalle de validité de [asin](#) .

Voir aussi [asinh](#) , [acos](#) et [atan](#) .

8.65.11 asinh() : Arc sinus hyperbolique

float **asinh** (float arg)

[asinh](#) retourne l'arc sinus hyperbolique de arg , c'est à dire la valeur dont le sinus hyperbolique est arg .

Note
Cette fonction n'est pas implémentée sous Windows.

Voir aussi [asin](#) , [acosh](#) et [atanh](#) .

8.65.12 atan() : arc tangent

float **atan** (float arg)

atan retourne l'arc tangent de arg (arg en radians). atan est la fonction inverse de tan , ce qui signifie que $a = \tan(\text{atan}(a))$ pour toute valeur qui soit dans l'intervalle de validité de atan .

Voir aussi atanh , asin et acos .

8.65.13 atan2() : Arc tangent de deux variables

float **atan2** (float y , float x)

atan2 retourne l'arc tangent de deux variables x et y . La formule est : " arc tangent (y / x) ", et les signes des arguments sont utilisés pour déterminer le quadrant du résultat.

atan2 retourne un résultat en radians, entre -PI et PI (inclus).

Voir aussi acos et atan .

8.65.14 atanh() : Arc tangeant hyperbolique

float **atanh** (float arg)

atanh retourne l'arc tangente hyperbolique de arg , c'est à dire la valeur dont la tangente hyperbolique est arg .

Note

Cette fonction n'est pas implémentée sous Windows.
--

Voir aussi atan , asinh et acosh .

8.65.15 base_convert() : Convertit un nombre entre des bases arbitraires

string **base_convert** (string number , int frombase , int tobase)

base_convert retourne une chaîne contenant l'argument number représenté dans la base tobase . La base de représentation de number est donnée par frombase . frombase et tobase doivent être compris entre 2 et 36 inclus. Les chiffres supérieurs à 10 des bases supérieures à 10 seront représentés par les lettres de A à Z, avec A = 10 et Z = 36.

Exemple avec <u>base_convert</u>

```
<?php
$hexadecimal = 'A37334';
echo base_convert($hexadecimal, 16, 2);
?>
```

L'exemple ci-dessus va afficher :

101000110111001100110100

Attention

<u>base_convert</u> perdra la précision sur les grands nombres dû à la propriété interne des types "double" et "float". Merci de regarder la section sur les <u>nombres à virgule flottante</u> dans ce manuel pour plus d'informations.
--

8.65.16 bindec() : Convertit de binaire en décimal

number **bindec** (string binary_string)

bindec retourne la conversion d'un nombre binaire représenté par la chaîne binary_string en décimal.

bindec convertit un nombre binaire en décimal. Le plus grand nombre convertible a 31 bits à 1, soit 2147483647 en décimal. Depuis PHP 4.1.0, cette fonction peut également convertir de très grands nombres. Elle retourne un nombre de type float dans ce cas.

Exemple avec <u>bindec</u>

<pre><?php echo bindec('110011') . "\n"; echo bindec('000110011') . "\n"; echo bindec('111'); ?></pre>

L'exemple ci-dessus va afficher :

<pre>51 51 7</pre>

Voir aussi decbin , octdec , hexdec et base_convert .

8.65.17 ceil() : Arrondit au nombre supérieur

float **ceil** (float value)

ceil retourne l'entier supérieur du nombre value . Utiliser ceil sur un entier ne sert à rien. La valeur retournée est un nombre à virgule flottante (float), car ces nombres peuvent être plus grands que les entiers.

Exemple avec <u>ceil</u>

<pre><?php echo ceil(4.3); // 5 echo ceil(9.999); // 10 ?></pre>
--

Voir aussi floor et round .

8.65.18 cos() : cosinus

float **cos** (float arg)

cos retourne le cosinus de arg (arg en radians).

Exemple avec cos

```
<?php
echo cos(M_PI); // -1
?>
```

Voir aussi [acos](#) , [sin](#) [tan](#) et [deg2rad](#) .

8.65.19 cosh() : Cosinus hyperbolic

float **cosh** (float arg)

cosh retourne le cosinus hyperbolique de arg , défini comme $(\exp(\arg) + \exp(-\arg))/2$.

Voir aussi [cos](#) , [acosh](#) , [sin](#) et [tan](#) .

8.65.20 decbin() : Convertit de décimal en binaire

string **decbin** (int number)

decbin retourne une chaîne contenant la représentation binaire de l'entier donné en argument. Le plus grand nombre pouvant être converti est 2147483647 en décimal, ce qui donne une série de 31 uns (1).

Exemple avec decbin

```
<?php
echo decbin(12) . "\n";
echo decbin(26);
?>
```

L'exemple ci-dessus va afficher :

```
1100
11010
```

Voir aussi [bindec](#) , [decoct](#) , [dechex](#) et [base_convert](#) .

8.65.21 dechex() : Convertit de décimal en hexadécimal

string **dechex** (int number)

dechex retourne une chaîne contenant la représentation hexadécimale du nombre number . Le nombre le plus grand qui puisse être converti est 4294967295 en décimal, ce qui donnera "ffffffff".

Exemple avec `dechex`

```
<?php
echo dechex(10) . "\n";
echo dechex(47);
?>
```

L'exemple ci-dessus va afficher :

```
a
2f
```

Voir aussi [hexdec](#) , [dechbin](#) , [decoct](#) et [base_convert](#) .

8.65.22 `decoct()` : Convertit de décimal en octal

string **decoct** (int number)

`decoct` retourne une chaîne contenant la représentation octale du nombre donné number . Le nombre le plus grand qui puisse être converti est 4294967295 en décimal, ce qui donnera "37777777777".

Exemple avec `decoct`

```
<?php
echo decoct(15) . "\n";
echo decoct(264);
?>
```

L'exemple ci-dessus va afficher :

```
17
410
```

Voir aussi [octdec](#) , [dechbin](#) , [dechex](#) et [base_convert](#) .

8.65.23 `deg2rad()` : Convertit un nombre de degrés en radians

float **deg2rad** (float number)

`deg2rad` convertit number de degrés en radians.

Exemple avec `deg2rad`

```
<?php

echo deg2rad(45); // 0.785398163397
var_dump(deg2rad(45) === M_PI_4); // bool(true)

?>
```

Voir aussi [rad2deg](#) .

8.65.24 exp() : Exponentiellefloat **exp** (float arg)exp retourne l'exponentielle de arg , c'est-à-dire e élevé à la puissance arg .**Note**

'e' est le logarithme naturel, ou approximativement 2.718282.

Exemple avec exp

```
<?php
echo exp(12) . "\n";
echo exp(5.7);
?>
```

L'exemple ci-dessus va afficher :

1.6275E+005

298.87

Voir aussi pow et log .**8.65.25 expm1() : Retourne le résultat de exp(number) - 1, calculé de manière précise, même si val est proche de 0**float **expm1** (float number)**Attention**

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

expm1 retourne l'équivalent de 'exp(number) - 1' calculé de tel sorte qu'il sera précis, même si la valeur de l'argument number est proche de 0 , un cas où l'expression 'exp (number) - 1' n'est pas précis, du fait de la soustraction de deux nombres quasi-égaux.

Note

Cette fonction n'est pas implémentée sous Windows.

Voir aussi log1p et exp .**8.65.26 floor() : Arrondit à l'entier inférieur**float **floor** (float value)

floor retourne l'entier inférieur du nombre value . La valeur retournée est un nombre à virgule flottante, (float) car ces nombres peuvent être plus grands que les entiers.

Exemple avec floor

```
<?php
echo floor(4.3); // 4
echo floor(9.999); // 9
```

```
?>
```

Voir aussi [ceil](#) et [round](#) .

8.65.27 fmod() : Retourne le reste de la division

float **fmod** (float x , float y)

fmod retourne le reste de la division de (x) par (y). Ce reste est un nombre à virgule flottante. Le reste est défini par : $x = i * y + r$, pour un entier i . Si y n'est pas nul, r a le même signe que x est une taille inférieure à y .

Exemple avec [fmod](#)

```
<?php
$x = 5.7;
$y = 1.3;
$r = fmod($x, $y);
// $r égale 0.5, car 4 * 1.3 + 0.5 = 5.7
?>
```

8.65.28 getrandmax() : Plus grande valeur aléatoire possible

int **getrandmax** (void)

[getrandmax](#) retourne la plus grande valeur aléatoire possible retournée par [rand](#) .

Voir aussi [rand](#) , [srand](#) et [mt_getrandmax](#) .

8.65.29 hexdec() : Convertit de hexadécimal en décimal

number **hexdec** (string hex_string)

[hexdec](#) retourne une chaîne contenant la représentation décimale du nombre hex_string . Le nombre le plus grand qui puisse être converti est 7fffffff en décimal, ce qui donne "2147483647". Depuis PHP 4.1.0, cette fonction peut également convertir de très grands nombres. Elle retourne un nombre de type float dans ce cas.

[hexdec](#) remplace tous les caractères non-hexadécimaux par des 0 . Et si les zéros de gauche sont ignorés, ceux de droite prennent le propre valeur.

Exemple avec [hexdec](#)

```
<?php
var_dump(hexdec("Hop comme ceci"));
var_dump(hexdec("0000c000e0cec0"));
var_dump(hexdec("c000e0cec0"));
// les deux affichent "int(14732992)"

var_dump(hexdec("aussi"));
var_dump(hexdec("a0000"));
// les deux affichent "int(655360)"
?>
```

Voir aussi [dechex](#) , [bindec](#) , [octdec](#) et [base_convert](#) .

8.65.30 **hypot()** : Calcul la longueur de l'hypoténuse d'un triangle à angle droit

float **hypot** (float *x* , float *y*)

hypot retourne la longueur de l'hypoténuse d'un triangle à angle droit qui a des cotés d'une longueur *x* et *y* ou bien la distance du point (*x* , *y*) depuis l'origine. Ceci est l'équivalent de $\sqrt{x*x + y*y}$.

8.65.31 **is_finite()** : Indique si un nombre est fini

bool **is_finite** (float *val*)

is_finite retourne TRUE si *val* est une valeur finie, c'est à dire une valeur qui peut être représentée par un nombre à virgule flottante sur cette plate-forme.

Voir aussi is_infinite et is_nan .

8.65.32 **is_infinite()** :

bool **is_infinite** (float *val*)

is_infinite retourne TRUE si *val* est infini (positif ou négatif), comme le résultat de $\log(0)$ ou une valeur qui est trop grande pour être représenté par un nombre à virgule flottante sur cette plate-forme.

Voir aussi is_finite et is_nan .

8.65.33 **is_nan()** : Indique si une valeur est un nombre

bool **is_nan** (float *val*)

is_nan retourne TRUE si *val* n'est pas un nombre, comme le résultat de $\text{acos}(1.01)$.

Voir aussi is_finite et is_infinite .

8.65.34 **lcg_value()** : Générateur de congruence combinée linéaire

float **lcg_value** (void)

lcg_value retourne un nombre pseudo-aléatoire, compris entre 0 et 1. lcg_value combine deux générateurs de congruence, de périodes respectives $2^{31} - 85$ et $2^{31} - 249$. La période de cette fonction est le produit de ces deux nombres premiers (soit $(2^{31} - 85) * (2^{31} - 249)$).

8.65.35 **log()** : Logarithme naturel (népérien)

float **log** (float *arg* , float *base*)

[log](#) retourne le logarithme naturel (ou népérien) de `arg` . Si le paramètre optionnel `base` est spécifié, [log](#) retourne alors le logarithme en base `base` .

Note

Le paramètre <code>base</code> a été ajouté en PHP version 4.3.0.

Vous pouvez toujours calculer le logarithme en base `b` d'un nombre `n` , en utilisant la formule ; $\log_b(n) = \log(n)/\log(b)$, où `log` est le logarithme naturel ou népérien.

Voir aussi [exp](#) .

8.65.36 `log10()` : logarithme en base 10

float **log10** (float `arg`)

[log10](#) retourne le logarithme en base 10 de `arg` .

Voir aussi [log](#)

8.65.37 `log1p()` : Retourne le résultat de `log(1 + number)`, calculé de manière précise, même si `val` est proche de zéro

float **log1p** (float `number`)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

[log1p](#) retourne `log(1 + number)` calculé de telle sorte qu'il sera précis même si la valeur de `number` est proche de 0 .

Note

Cette fonction n'est pas implémentée sous Windows.
--

Voir aussi [expm1](#) et [log](#) .

8.65.38 `max()` : La plus grande valeur

mixed **max** (number `arg1` , number `arg2` , *number* ...) mixed **max** (array `numbers`)

[max](#) retourne la plus grande valeur numérique parmi les valeurs passées en paramètre.

Si le premier paramètre est un tableau, [max](#) retourne la plus grande valeur de ce tableau. Si le premier paramètre est un entier, une chaîne ou un nombre à virgule flottante (float), [max](#) requiert au moins deux paramètres, et retournera alors le plus grand d'entre eux. Le nombre d'arguments est alors illimité.

Note

PHP évaluera une <u>chaîne de caractères</u> non-numérique en tant que 0 , mais continuera de retourner une chaîne de caractères s'il s'aperçoit qu'elle a une valeur numérique supérieure. Si de multiples arguments sont évalués à 0 , max retournera un 0 numérique s'il est fourni, sinon, la

chaîne de caractères alphabétiquement supérieure sera retournée.

Exemple d'utilisation de max

```
<?php
echo max(1, 3, 5, 6, 7); // 7
echo max(array(2, 4, 5)); // 5

echo max(0, 'bonjour'); // 0
echo max('bonjour', 0); // bonjour
echo max(-1, 'bonjour'); // bonjour

// Avec plusieurs tableaux, min() fait les comparaisons de gauche à droite
// dans notre exemple : 2 == 2, mais 4 < 5
$val = max(array(2, 4, 8), array(2, 5, 7)); // array(2, 5, 7)

// Si un mélange de tableau et de scalaires sont fournis,
// le tableau est toujours retourné, car il est considéré comme le plus grand
$val = max('string', array(2, 5, 7), 42); // array(2, 5, 7)
?>
```

Voir aussi min et count .

8.65.39 min() : La plus petite valeur

mixed **min** (number arg1 , number arg2 , *number* ...) mixed **min** (array numbers)

min retourne la plus petite valeur numérique parmi les valeurs passées en paramètres.

Si le premier et le seul paramètre est un tableau, min retournera la plus petite valeur contenue dans le tableau. Si le premier paramètre est un entier, une chaîne ou un nombre décimal, vous devez fournir au moins deux paramètres et min retournera la plus petite de ces valeurs. Vous pouvez comparer un nombre illimité de valeurs.

Note

PHP évaluera une chaîne de caractères non-numérique en tant que 0 , mais continuera de retourner une chaîne de caractères s'il s'aperçoit qu'elle a une valeur numérique supérieure. Si de multiples arguments sont évalués à 0 , max retournera un 0 numérique s'il est fourni, sinon, la chaîne de caractères alphabétiquement supérieure sera retournée.

Exemples avec min

```
<?php
echo min(2, 3, 1, 6, 7); // 1
echo min(array(2, 4, 5)); // 2

echo min(0, 'bonjour'); // 0
echo min('bonjour', 0); // bonjour
echo min('bonjour', -1); // -1

// Avec plusieurs tableaux, min() fait les comparaisons de gauche à droite
// dans notre exemple : 2 == 2, mais 4 < 5
$val = min(array(2, 4, 8), array(2, 5, 1)); // array(2, 4, 8)

// Si un mélange de tableau et de scalaires sont fournis,
// le tableau n'est jamais retourné, car il est considéré comme le plus grand
$val = min('chaîne', array(2, 5, 7), 42); // chaîne
?>
```

Voir aussi max et count .

8.65.40 mt_getrandmax() : La plus grand valeur aléatoire possible

int **mt_getrandmax** (void)

mt_getrandmax retourne la plus grand valeur aléatoire possible que peut retourner mt_rand .

Voir aussi mt_rand , mt_srand et getrandmax .

8.65.41 mt_rand() : Génère une valeur aléatoire (meilleure méthode)

int **mt_rand** (int *min* , int *max*)

De nombreux générateurs de nombres aléatoires provenant de vieilles bibliothèques libcs ont des comportements douteux et sont très lents. Par défaut, PHP utilise le générateur de nombres aléatoires de libc avec la fonction rand . mt_rand est une fonction de remplacement, pour cette dernière. Elle utilise un générateur de nombres aléatoire de caractéristique connue, le " Mersenne Twister " qui est 4 fois plus rapide que la fonction standard libc. La "Homepage of the Mersenne Twister " est <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html> .

Appelée sans les arguments optionnels min et max , mt_rand retourne un nombre pseudo-aléatoire, entre 0 et RAND_MAX . Pour obtenir un nombre entre 5 et 15 inclus, il faut utiliser mt_rand(5,15) .

Exemple avec mt_rand

```
<?php
echo mt_rand() . "\n";
echo mt_rand() . "\n";

echo mt_rand(5, 15);
?>
```

L'exemple ci-dessus va afficher quelque chose de similaire à :

```
1604716014
1478613278
6
```

Note

Depuis PHP 4.2.0, vous n'avez plus besoin d'initialiser le générateur de nombres aléatoires avec srand ou mt_srand car c'est fait automatiquement.

Note

Dans les versions antérieures à la 3.0.7, la signification du paramètre max était "longueur". Pour avoir le même résultat, il faut utiliser mt_rand (5, 11) pour obtenir un nombre aléatoire entre 5 et 15.

Voir aussi mt_srand , mt_getrandmax et rand .

8.65.42 mt_srand() : Initialise une valeur aléatoire (meilleure méthode)

void **mt_srand** (int *seed*)

mt_srand initialise une meilleure valeur aléatoire avec seed . Depuis PHP 4.2.0, le paramètre seed est devenu optionnel et est généré par PHP s'il n'est pas renseigné.

Exemple avec `mt_srand`

```
<?php
// initialisation avec des microsecondes
function make_seed()
{
    list($usec, $sec) = explode(' ', microtime());
    return (float) $sec + ((float) $usec * 100000);
}
mt_srand(make_seed());
$randval = mt_rand();

?>
```

Note

Depuis PHP 4.2.0, vous n'avez plus besoin d'initialiser le générateur de nombres aléatoires avec `srand` ou `mt_srand` car c'est fait automatiquement.

Voir aussi `mt_rand` , `mt_getrandmax` et `srand` .

8.65.43 `octdec()` : Conversion d'octal en décimal

number `octdec` (string octal_string)

`octdec` retourne une chaîne contenant la représentation décimale du nombre octal_string . Le nombre le plus grand qui puisse être converti est 17777777777 en décimal, ce qui donnera "2147483647". Depuis PHP 4.1.0, cette fonction peut également convertir de très grands nombres. Elle retourne un nombre de type float dans ce cas.

Exemple avec `octdec`

```
<?php
echo octdec('77') . "\n";
echo octdec(decoct(45));
?>
```

L'exemple ci-dessus va afficher :

```
63
45
```

Voir aussi `decoct` , `bindec` , `hexdec` et `base_convert` .

8.65.44 `pi()` : Retourne la valeur de pi

float `pi` (void)

`pi` retourne la valeur de pi. La valeur retournée est de type float , et est à la précision indiquée par la directive `precision` , indiquée dans le fichier php.ini , qui vaut par défaut 14 . De plus, vous pouvez utiliser la constante `M_PI` , qui retourne un résultat identique à la fonction `pi` .

Exemple avec `pi`

```
<?php
echo pi(); // 3.1415926535898
echo M_PI; // 3.1415926535898
```

```
?>
```

8.65.45 pow() : Expression exponentielle

number **pow** (number base , number exp)

pow retourne base élevé à la puissance exp . Si possible, pow retourne un entier .

Si le calcul ne peut être fait, une alerte sera affichée et pow retournera FALSE . Depuis PHP 4.2.0, pow n'affiche plus aucun warning.

Note
PHP ne peut pas gérer une valeur négative de base .
Quelques exemples avec <u>pow</u>
<pre><?php var_dump(pow(2, 8)); // int(256) echo pow(-1, 20); // 1 echo pow(0, 0); // 1 echo pow(-1, 5.5); // error ?></pre>
Attention
En PHP 4.0.6 plus ancien, <u>pow</u> retournait toujours un nombre à virgule flottante (float), et n'affichait pas d'alerte. Si le calcul est impossible (racine d'un nombre négatif, par exemple), <u>pow</u> retournait NAN .

Voir aussi exp , sqrt , bcpow et gmp_pow .

8.65.46 rad2deg() : Conversion de radians en degrés

float **rad2deg** (float number)

rad2deg convertit number (supposé en radians) en degrés.

Exemple avec <u>rad2deg</u>
<pre><?php echo rad2deg(M_PI_4); // 45 ?></pre>

Voir aussi deg2rad .

8.65.47 rand() : Génère une valeur aléatoire

int **rand** (int min , int max)

Appelée sans les options min et max , rand retourne un nombre pseudo-aléatoire entre 0 et RAND_MAX . Si vous voulez un nombre aléatoire entre 5 et 15 (inclus), par exemple, utilisez rand (5, 15) .

Exemple avec `rand`

```
<?php
echo rand() . "\n";
echo rand() . "\n";

echo rand(5, 15);
?>
```

L'exemple ci-dessus va afficher quelque chose de similaire à :

```
7771
22264
11
```

Note

Sur quelques plate-formes (comme Windows par exemple), `RAND_MAX` est limité à 32768. Si vous désirez une limite supérieure à 32768, en spécifiant min et max , vous serez autorisés à utiliser un intervalle plus grand que `RAND_MAX`, ou bien, utilisez la fonction `mt_rand` à la place.

Note

Depuis PHP 4.2.0, vous n'avez plus besoin d'initialiser le générateur de nombres aléatoires avec `srand` ou `mt_srand` car c'est fait automatiquement.

Note

Dans les versions antérieures à la 3.0.7 la signification du paramètre max était longueur. Pour avoir le même résultat, il faut utiliser `mt_rand (5, 11)` pour obtenir un nombre aléatoire entre 5 et 15.

Voir aussi `srand` , `mt_rand` et `getrandmax` .

8.65.48 `round()` : Arrondi

float **round** (float *val* , int *precision*)

`round` retourne la valeur arrondie de *val* à la précision *precision* (nombre de chiffres après la virgule). Le paramètre *precision* peut être négatif ou NULL : c'est sa valeur par défaut.

Exemple avec `round`

```
<?php
echo round(3.4);           // 3
echo round(3.5);           // 4
echo round(3.6);           // 4
echo round(3.6, 0);        // 4
echo round(1.95583, 2);    // 1.96
echo round(1241757, -3);   // 1242000
echo round(5.045, 2);      // 5.05
echo round(5.055, 2);      // 5.06
?>
```

Note

PHP ne gère pas correctement les chaînes telles que "12 300,2" , par défaut. Reportez-vous à [la conversion de chaînes](#) .

Note

Le paramètre *precision* a été introduit depuis PHP 4.

Voir aussi [ceil](#) , [floor](#) et [number_format](#) .

8.65.49 sin() : Sinus

float **sin** (float arg)

[sin](#) retourne le sinus de arg (arg in radians).

Exemple avec [sin](#)

```
<?php
// La précision dépend de la directive precision
echo sin(deg2rad(60)); // 0.866025403 ...
echo sin(60);         // -0.304810621 ...
?>
```

Voir aussi [asin](#) , [cos](#) , [tan](#) et [deg2rad](#) .

8.65.50 sinh() : Sinyus hyperbolique

float **sinh** (float arg)

[sinh](#) retourne le sinus hyperbolique de arg , défini comme $(\exp(\arg) - \exp(-\arg))/2$.

Voir aussi [sin](#) , [asinh](#) , [cos](#) et [tan](#) .

8.65.51 sqrt() : Racine carrée

float **sqrt** (float arg)

[sqrt](#) retourne la racine carrée de arg .

Exemple avec [sqrt](#)

```
<?php
// La précision dépend de votre directive precision
echo sqrt(9); // 3
echo sqrt(10); // 3.16227766 ...
?>
```

Voir aussi [pow](#) .

8.65.52 srand() : Initialise le générateur de nombres aléatoires

void **srand** (int seed)

[srand](#) initialise le générateur de nombres aléatoires avec seed . Depuis PHP 4.2.0, le paramètre seed est devenu optionnel et est généré par PHP s'il n'est pas renseigné.

Exemple avec [srand](#)

```
<?php
```

```
// initialise avec les microsecondes
function make_seed()
{
    list($usec, $sec) = explode(' ', microtime());
    return (float) $sec + ((float) $usec * 100000);
}
srand(make_seed());
$randval = rand();
?>
```

Note

Depuis PHP 4.2.0, vous n'avez plus besoin d'initialiser le générateur de nombres aléatoires avec `srand` ou `mt_srand` car c'est fait automatiquement.

Voir aussi [rand](#) , [getrandmax](#) et [mt_srand](#) .

8.65.53 tan() : Tangente

float **tan** (float arg)

[tan](#) retourne la tangente de arg (arg en radians).

Exemple avec [tan](#)

```
<?php
echo tan(M_PI_4); // 1
?>
```

Voir aussi [atan](#) , [sin](#) , [cos](#) et [deg2rad](#) .

8.65.54 tanh() : Tangente hyperbolique

float **tanh** (float arg)

[tanh](#) retourne la tangente hyperbolique de arg , définie comme $\sinh(\arg)/\cosh(\arg)$.

Voir aussi [tan](#) , [atanh](#) , [sin](#) et [cos](#) .

8.66 MaxDB PHP Extension

8.66.1 Introduction

L'extension MaxDB de PHP vous permet d'accéder aux fonctionnalités fournies par MaxDB 7.5.0 et suivant. Plus d'informations sur le serveur de base de données MaxDB peuvent être trouvées sur <http://www.mysql.com/products/maxdb/>.

L'extension MaxDB est compatible avec l'extension mysqli de MySQL. Il y a simplement des différences mineurs dans le comportement de certaines fonctions, liées au comportement des bibliothèques sous-jacentes.

Les différences notables sont :

- [maxdb_character_set_name](#) : retourne uniquement de l'ASCII ou unicode
- [maxdb_disable_reads_from_master](#) : fonction expérimentale
- [maxdb_disable_rpl_parse](#) : fonction expérimentale
- [maxdb_dump_debug_info](#) : fonction expérimentale
- [maxdb_embedded_connect](#) : fonction expérimentale
- [maxdb_enable_reads_from_master](#) : fonction expérimentale
- [maxdb_enable_rpl_parse](#) : fonction expérimentale
- [maxdb_get_client_info](#) : version différente
- [maxdb_get_client_version](#) : version différente
- [maxdb_get_host_info](#) : retourne localhost ou le nom d'hôte
- [maxdb_get_server_info](#) : version différente
- [maxdb_get_server_version](#) : version différente
- [maxdb_kill](#) : déconnecte la session
- [maxdb_master_query](#) : fonction expérimentale
- [maxdb_more_results](#) : la fonction retourne toujours FALSE
- [maxdb_multi_query](#) : ne peut pas gérer des commandes SQL multiples
- [maxdb_next_result](#) : la fonction retourne toujours FALSE
- [maxdb_options](#) : les options sont différentes
- [maxdb_report](#) : les modes de rapports sont différents
- [maxdb_rpl_parse_enabled](#) : fonction expérimentale
- [maxdb_rpl_probe](#) : fonction expérimentale
- [maxdb_rpl_query_type](#) : fonction expérimentale
- [maxdb_send_query](#) : fonction expérimentale
- [maxdb_server_end](#) : fonction expérimentale
- [maxdb_server_init](#) : fonction expérimentale
- [maxdb_ssl_set](#) : fonction expérimentale
- [maxdb_stat](#) : système de status différent
- [maxdb_stmt_store_result](#) : sans effet
- [maxdb_store_result](#) : sans effet

La documentation sur MaxDB est disponible sur <http://dev.mysql.com/doc/maxdb/>.

8.66.2 Pré-requis

Pour pouvoir utiliser ces fonctions, vous devez compiler PHP avec le support MaxDB. Additionnellement, vous devez avoir de disponible la bibliothèque MaxDB SQLDBC pour accéder au serveur MaxDB.

La documentation sur MaxDB SQLDBC peut être trouvée sur <http://dev.mysql.com/doc/maxdb/>.

Téléchargez le paquet MaxDB SQLDBC sur <http://dev.mysql.com/downloads/maxdb/clients.html>.

8.66.3 Installation

En utilisant l'option de configuration `--with-maxdb[=DIR]`, vous permettez à PHP d'accéder à une base de données MaxDB. [DIR] pointe vers le dossier contenant le paquet d'installation MaxDB SQLDBC.

Les utilisateurs de Windows doivent activer la bibliothèque `php_maxdb.dll` dans le `php.ini`.

8.66.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable	Historique
<code>maxdb.default_host</code>	NULL	PHP_INI_ALL	
<code>maxdb.default_db</code>	NULL	PHP_INI_ALL	
<code>maxdb.default_user</code>	NULL	PHP_INI_ALL	
<code>maxdb.default_pw</code>	NULL	PHP_INI_ALL	
<code>maxdb.long_readlen</code>	"200"	PHP_INI_ALL	

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

`maxdb.default_host` string

L'hôte par défaut à utiliser lors de la connexion au serveur de base de données si aucun hôte n'est spécifié.

`maxdb.default_db` string

La base de données par défaut à utiliser lors de la connexion si aucune autre base de données n'est spécifiée.

`maxdb.default_user` string

Le nom d'utilisateur par défaut à utiliser lors de la connexion au serveur de base de données si aucun autre nom n'est spécifié.

`maxdb.default_pw` string

Le mot de passe par défaut à utiliser lorsqu'un utilisateur se connecte à un serveur sans spécifier de mot de passe.

`maxdb.long_readlen` integer

La longueur maximale d'octets par défaut qui sont transférés au client si un gros volume de données est récupéré depuis un serveur de base de données MaxDB.

8.66.5 Classes pré-définies

8.66.5.1 maxdb

Représente une connexion entre PHP et une base MaxDB.

8.66.5.1.1 Constructeur

- maxdb : constructeur

8.66.5.1.2 Méthodes

- autocommit : active ou pas l'auto-archivage
- change_user : change l'utilisateur de la base de données
- character_set_name : retourne le jeu de caractères par défaut de la connexion
- close : ferme une connexion ouverte
- commit : valide la transaction courante
- connect : ouvre une nouvelle connexion
- debug : effectue des opérations de débogage
- dump_debug_info : affiche des informations de débogage
- get_client_info : retourne la version du client
- get_host_info : retourne le type de connexion
- get_server_info : retourne les informations sur le serveur maxDB
- get_server_version : retourne la version du serveur
- init : initialise un objet maxdb
- info : lit les informations sur la dernière requête exécutée
- kill : tente de terminer un thread MaxDB
- multi_query : effectue une commande multiple
- more_results : vérifie si d'autres résultats existent dans la commande multiple courante
- next_result : lit le prochain résultat de la commande multiple courante
- options : configure les options
- ping : pingue le serveur et se reconnecte si besoin
- prepare : prépare une requête SQL
- query : exécute une requête
- real_connect : ouvre une connexion au serveur MaxDB
- escape_string : protège les caractères spéciaux dans une chaîne pour l'utiliser dans une requête, en tenant compte du jeu de caractères courant
- rollback : annule la transaction courante
- select_db : sélectionne la base de données par défaut
- ssl_set : configure les paramètres ssl
- stat : lit le statut courant du système
- stmt_init : initialise une commande à utiliser avec maxdb_stmt_prepare
- store_result : transfère un résultat de la dernière requête
- use_result : transfère un résultat non bufferisé de la dernière requête
- thread-safe : indique si la sécurité thread est active ou pas

8.66.5.1.3 Propriétés

- affected_rows : lit le nombre de lignes affectées par la dernière opération MaxDB
- client_info : retourne la version du client MaxDB sous forme de chaîne
- client_version : retourne la version du client MaxDB sous forme d'entier
- errno : retourne le code d'erreur du dernier appel de fonction
- error : retourne le message d'erreur de la plus récente fonction
- field_count : return le nombre de colonne de la dernière requête
- host_info : return une chaîne représentant le type de connexion
- info : lit les informations sur la dernière requête exécutée
- insert-id : retourne l'identifiant auto-généré par la dernière requête
- protocol_version : retourne la version du protocole MaxDB utilisé
- sqlstate : retourne une chaîne représentant le code d'erreur SQLSTATE

- thread_id : retourne l'identifiant du thread de la connexion courante
- warning-count : retourne le nombre d'alertes générées durant la dernière commande SQL

8.66.5.2 maxdb_stmt

Représente une commande préparée.

8.66.5.2.1 Méthodes

- bind_param : lie les variables à la commande préparée
- bind_result : lie les variables à la commande préparée, pour stockage des résultats
- close : détruit la commande préparée
- data_seek : place le pointeur à une position arbitraire du résultat
- execute : exécute une commande préparée
- fetch : lit les résultats de la commande préparée dans les variables liées
- free_result : libère les ressources de la mémoire
- result_metadata : lit les méta-données d'un résultat de commande préparée
- prepare : prépare une commande préparée
- send_long_data : envoie les données par paquet
- close_long_data : termine l'envoi de grandes données
- reset : remet à zéro une commande préparée
- store_result : stocke tout le résultat d'une commande préparée

8.66.5.2.2 Propriétés

- affected_rows : retourne le nombre de lignes affectées par la dernière commande préparée
- errno : retourne le code d'erreur de la dernière commande préparée
- errno : retourne le message d'erreur de la dernière commande préparée
- param_count : retourne le nombre de paramètres pour une commande préparée
- sqlstate : retourne une chaîne contenant le code d'erreur SQLSTATE

8.66.5.3 maxdb_result

Représente le jeu de résultat obtenu après exécution d'une requête.

8.66.5.3.1 Méthodes

- close : détruit le jeu de résultat
- data_seek : déplace le pointeur de ligne
- fetch_field : lit les informations de colonnes d'un résultat
- fetch_fields : lit les informations de toutes les colonnes d'un résultat
- fetch_field_direct : lit les informations sur une colonne spécifique
- fetch_array : lit la ligne de résultat sous forme de tableau associatif, numérique ou les deux.
- fetch_assoc : lit une ligne sous forme de tableau associatif
- fetch_object : lit un résultat sous forme d'objet
- fetch_row : lit un résultat sous forme de tableau énuméré
- close : libère les ressources
- field_seek : place le pointeur à une position donnée

8.66.5.3.2 Propriétés

- current_field : retourne le pointeur courant
- field_count : retourne le nombre de champs dans le résultat
- lengths : retourne un tableau avec les tailles des colonnes
- num_rows : retourne le nombre de ligne du résultat

8.66.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Les constantes suivantes à utiliser avec maxdb_options sont définies. Pour plus de descriptions sur ces constantes, lisez <http://dev.mysql.com/doc/maxdb/>.

Constante	Description
MAXDB_COMPNAME	Le nom du composant utilisé pour initialiser l'environnement d'exécution SQLDBC.
MAXDB_APPLICATION	L'application à connecter à la base de données.
MAXDB_APPVERSION	La version de l'application.
MAXDB_SQLMODE	Le mode SQL.
MAXDB_UNICODE	TRUE , si la connexion est un client unicode (UCS2) ou FALSE sinon.
MAXDB_TIMEOUT	Le délai maximal d'inactivité après lequel la connexion à la base de données est close par le système.
MAXDB_ISOLATIONLEVEL	Spécifie si oui ou non et comment les verrous partagés ainsi que les verrous exclusifs sont implicitement demandés ou libérés.
MAXDB_PACKETCOUNT	Le nombre de paquets de requêtes différents utilisé pour la connexion.
MAXDB_STATEMENTCACHESIZE	Le nombre de requêtes préparées à être mises en cache pour la connexion pour être réutilisées.
MAXDB_CURSORPREFIX	Le préfixe à utiliser pour la table de résultats qui sont automatiquement nommées.

La fonction maxdb_fetch_array utilise une constante pour les différents types de tableaux de résultats. Les constantes suivantes sont définies :

Constante	Description
MAXDB_ASSOC	Les colonnes sont retournées dans un tableau ayant comme index les noms des champs.
MAXDB_ASSOC_UPPER	Les colonnes sont retournées dans un tableau ayant comme index les noms des champs en majuscule.
MAXDB_ASSOC_LOWER	Les colonnes sont retournées dans un tableau ayant comme index les noms des champs en minuscule.
MAXDB_BOTH	Les colonnes sont retournées dans un tableau ayant des index numériques ainsi que les noms des champs en tant qu'index.
MAXDB_NUM	Les colonnes sont retournées dans un tableau ayant un index numérique correspondant à un champs. Cet index commence à 0, le premier champs du jeu de résultats.

8.66.7 Exemples

Tous les exemples dans la documentation PHP de MaxDB utilise la base de données de démonstration HETOLDB depuis MaxDB. Plus d'informations concernant cette base de données peuvent être trouvées sur

<http://dev.mysql.com/doc/maxdb/en/98/11b83fa6b33c17e10000000a114084/frameset.htm> .

Cet exemple simple montre comment se connecter, exécuter une requête, afficher les lignes résultants et se déconnecter de la base de données MaxDB.

Exemple global de l'extension MaxDB

```
<?php
/* Connexion et sélection de la base de données */
$link = maxdb_connect("maxdb_host", "maxdb_user", "maxdb_password")
or die("Connexion impossible : " . maxdb_connect_error());
echo "Connexion avec succès";
maxdb_select_db("my_database") or die("Impossible de sélectionner la base de données");

/* Exécution d'une requête SQL */
$query = "SELECT * FROM my_table";
$result = maxdb_query($link, $query) or die("Query failed : " . maxdb_error());

/* Affichage des résultats en HTML */
echo "<table>\n";
while ($line = maxdb_fetch_array($result, MAXDB_ASSOC)) {
    echo "\t<tr>\n";
    foreach ($line as $col_value) {
        echo "\t\t<td>$col_value</td>\n";
    }
    echo "\t</tr>\n";
}
echo "</table>\n";

/* Libération des ressources */
maxdb_free_result($result);

/* Fermeture de la connexion */
maxdb_close($link);
?>
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Classes pré-définies](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [maxdb_affected_rows](#)
- [maxdb_autocommit](#)
- [maxdb_bind_param](#)
- [maxdb_bind_result](#)
- [maxdb_change_user](#)
- [maxdb_character_set_name](#)
- [maxdb_client_encoding](#)
- [maxdb_close_long_data](#)
- [maxdb_close](#)
- [maxdb_commit](#)
- [maxdb_connect_errno](#)

- [maxdb_connect_error](#)
- [maxdb_connect](#)
- [maxdb_data_seek](#)
- [maxdb_debug](#)
- [maxdb_disable_reads_from_master](#)
- [maxdb_disable_rpl_parse](#)
- [maxdb_dump_debug_info](#)
- [maxdb_embedded_connect](#)
- [maxdb_enable_reads_from_master](#)
- [maxdb_enable_rpl_parse](#)
- [maxdb_errno](#)
- [maxdb_error](#)
- [maxdb_escape_string](#)
- [maxdb_execute](#)
- [maxdb_fetch_array](#)
- [maxdb_fetch_assoc](#)
- [maxdb_fetch_field_direct](#)
- [maxdb_fetch_field](#)
- [maxdb_fetch_fields](#)
- [maxdb_fetch_lengths](#)
- [maxdb_fetch_object](#)
- [maxdb_fetch_row](#)
- [maxdb_fetch](#)
- [maxdb_field_count](#)
- [maxdb_field_seek](#)
- [maxdb_field_tell](#)
- [maxdb_free_result](#)
- [maxdb_get_client_info](#)
- [maxdb_get_client_version](#)
- [maxdb_get_host_info](#)
- [maxdb_get_metadata](#)
- [maxdb_get_proto_info](#)
- [maxdb_get_server_info](#)
- [maxdb_get_server_version](#)
- [maxdb_info](#)
- [maxdb_init](#)
- [maxdb_insert_id](#)
- [maxdb_kill](#)
- [maxdb_master_query](#)
- [maxdb_more_results](#)
- [maxdb_multi_query](#)
- [maxdb_next_result](#)
- [maxdb_num_fields](#)
- [maxdb_num_rows](#)
- [maxdb_options](#)
- [maxdb_param_count](#)
- [maxdb_ping](#)
- [maxdb_prepare](#)
- [maxdb_query](#)
- [maxdb_real_connect](#)
- [maxdb_real_escape_string](#)
- [maxdb_real_query](#)
- [maxdb_report](#)
- [maxdb_rollback](#)

- [maxdb_rpl_parse_enabled](#)
- [maxdb_rpl_probe](#)
- [maxdb_rpl_query_type](#)
- [maxdb_select_db](#)
- [maxdb_send_long_data](#)
- [maxdb_send_query](#)
- [maxdb_server_end](#)
- [maxdb_server_init](#)
- [maxdb_set_opt](#)
- [maxdb_sqlstate](#)
- [maxdb_ssl_set](#)
- [maxdb_stat](#)
- [maxdb_stmt_affected_rows](#)
- [maxdb_stmt_bind_param](#)
- [maxdb_stmt_bind_result](#)
- [maxdb_stmt_close_long_data](#)
- [maxdb_stmt_close](#)
- [maxdb_stmt_data_seek](#)
- [maxdb_stmt_errno](#)
- [maxdb_stmt_error](#)
- [maxdb_stmt_execute](#)
- [maxdb_stmt_fetch](#)
- [maxdb_stmt_free_result](#)
- [maxdb_stmt_init](#)
- [maxdb_stmt_num_rows](#)
- [maxdb_stmt_param_count](#)
- [maxdb_stmt_prepare](#)
- [maxdb_stmt_reset](#)
- [maxdb_stmt_result_metadata](#)
- [maxdb_stmt_send_long_data](#)
- [maxdb_stmt_sqlstate](#)
- [maxdb_stmt_store_result](#)
- [maxdb_store_result](#)
- [maxdb_thread_id](#)
- [maxdb_thread_safe](#)
- [maxdb_use_result](#)
- [maxdb_warning_count](#)

8.66.9 **maxdb_autocommit()** : **maxdb->auto_commit()**

Style procédural

bool **maxdb_autocommit** (resource link , bool mode)

Style orienté objet (méthode)

maxdb bool **auto_commit** (bool mode)

[maxdb_autocommit](#) est utilisé pour activer ou désactiver le mode auto-commit sur les requêtes pour la connexion à la base de données représentée par la ressource link .

8.66.10 **maxdb_bind_param()** : Alias de **maxdb_stmt_bind_param**

Cette fonction est un alias de : maxdb_stmt_bind_param

Cet alias est déprécié et n'existe que pour des raisons de compatibilité. Il est recommandé de ne pas utiliser cette fonction car elle peut être supprimée de PHP un jour.

8.66.11 **maxdb_bind_result()** : Alias de **maxdb_stmt_bind_result**

Cette fonction est un alias de : maxdb_stmt_bind_result .

Cet alias est déprécié et n'existe que pour des raisons de compatibilité. Il est recommandé de ne pas utiliser cette fonction car elle peut être supprimée de PHP un jour.

8.66.12 **maxdb_change_user()** : **maxdb->change_user()**

Style procédural

bool **maxdb_change_user** (resource link , string user , string password , string database)

Style orienté objet (méthode)

maxdb bool **change_user** (string user , string password , string database)

maxdb_change_user est utilisé pour modifier l'utilisateur de la connexion à la base de données spécifiée par le paramètre link et pour définir la base de données courant à celle spécifiée par le paramètre database .

Pour modifier avec succès un utilisateur, les paramètres username et password doivent être fournis et cet utilisateur doit avoir des permissions suffisantes pour accéder à la base de données désirées. Si pour une raison quelconque l'identification échoue, l'utilisateur courant sera conservé.

Note
L'utilisation de cette commande fera que la connexion courante à la base de données sera renouvelée, comme si l'on avait fait une nouvelle connexion, indépendamment du succès de l'opération. Cette réinitialisation effectue un rollback sur toutes les transactions actives, ferme toutes les tables temporaires et déverrouille toutes les tables verrouillées.

8.66.13 **maxdb_character_set_name()** : **maxdb->character_set_name()**

Style procédural

string **maxdb_character_set_name** (resource link)

Style orienté objet (méthode)

maxdb string **character_set_name** ()

Retourne le jeu de caractères courant défini pour la connexion à la base de données spécifiée par le paramètre link .

8.66.14 **maxdb_client_encoding()** : Alias de **maxdb_character_set_name**

Cette fonction est un alias de : maxdb_character_set_name .

Cet alias est déprécié et n'existe que pour des raisons de compatibilité. Il est recommandé de ne pas utiliser cette fonction car elle peut être supprimée de PHP un jour.

8.66.15 **maxdb_close_long_data()** : **maxdb->close_long_data()**

Cette fonction est un alias de : maxdb_stmt_close_long_data .

Cet alias est déprécié et n'existe que pour des raisons de compatibilité. Il est recommandé de ne pas utiliser cette fonction car elle peut être supprimée de PHP un jour.

8.66.16 **maxdb_close()** : **maxdb->close()**

Style procédural

bool **maxdb_close** (resource link)

Style orienté objet (méthode)

maxdb bool **close** ()

maxdb_close ferme une connexion à une base de données spécifiée par le paramètre link .

8.66.17 **maxdb_commit()** : **maxdb->commit()**

Style procédural

bool **maxdb_commit** (resource link)

Style orienté objet (méthode)

maxdb bool **commit** ()

maxdb_commit valide la transaction courante pour la connexion à la base de données spécifiée par le paramètre link .

8.66.18 **maxdb_connect_errno()** : Retourne le code erreur du dernier appel à la connexion

int **maxdb_connect_errno** ()

maxdb_connect_errno retourne le dernier code erreur du dernier appel à maxdb_connect . Si aucune erreur n'est survenue, cette fonction retournera zéro.

8.66.19 `maxdb_connect_error()` : Retourne une chaîne descriptive de la dernière erreur

string `maxdb_connect_error` ()

`maxdb_connect_error` est identique à la fonction `maxdb_connect_errno`, excepté qu'au lieu de retourner un code erreur, `maxdb_connect_error` retourne une chaîne représentative de la dernière erreur survenue lors du dernier appel à `maxdb_connect`. Si aucune erreur n'est survenue, cette fonction retournera une chaîne vide.

8.66.20 `maxdb_connect()` : `maxdb()`

Style procédural

resource `maxdb_connect` (*string host* , *string username* , *string passwd* , *string dbname* , *int port* , *string socket*)

Style orienté objet (méthode)

`maxdb` **__construct** *string host string username string passwd string dbname int port string socket*

`maxdb_connect` tente d'ouvrir une connexion sur un serveur MaxDB fonctionnant sur l'hôte `host` qui peut être soit un nom d'hôte, soit une adresse IP. Passer la chaîne "localhost" à ce paramètre, l'hôte local sera assumé. En cas de succès, `maxdb_connect` retournera une ressource représentant la connexion à la base de données, ou FALSE si une erreur survient.

Les paramètres `username` et `password` spécifient le nom de l'utilisateur ainsi que le mot de passe utilisés pour la connexion au serveur MaxDB. Si le mot de passe n'est pas fourni (la valeur NULL est passée), le serveur MaxDB tentera d'identifier l'utilisateur encore une fois en utilisant la valeur de l'option de configuration `maxdb.default_pw` du `php.ini`.

Le paramètre `dbname`, si fourni, spécifie la base de données par défaut à utiliser lors de l'exécution de requêtes. S'il n'est pas fourni, l'option de configuration `maxdb.default_db` du `php.ini` est utilisée.

Les paramètres `port` et `socket` sont ignorés par le serveur MaxDB.

8.66.21 `maxdb_data_seek()` : `result->data_seek()`

Style procédural

bool `maxdb_data_seek` (resource `result` , int `offset`)

Style orienté objet (méthode)

result bool `data_seek` (int `offset`)

`maxdb_data_seek` se déplace sur un pointeur de résultat arbitraire spécifié par `offset` dans le jeu de résultats représenté par `result`. Le paramètre `offset` doit être compris entre zéro le nombre total de lignes moins un (0.. `maxdb_num_rows` - 1).

8.66.22 maxdb_debug() : Effectue des opérations de débogage

void **maxdb_debug** (string debug)

maxdb_debug peut être utilisé pour tracer la connexion SQLDBC. Les chaînes suivantes peuvent être utilisées en tant que paramètre de la fonction maxdb_debug :

- TRACE SHORT ON|OFF - Active/Désactive l'appel à la méthode de traçage.
- TRACE LONG ON|OFF - Active/Désactive l'argument de la méthode et les détails de la trace de débogage.
- TRACE PACKET ON|OFF|<size> - Active/Désactive le paquet de trace, limitant la taille de l'objet de traçage au nombre d'octets spécifiés, ou 1000 si aucune taille n'est spécifiée.
- TRACE SQL ON|OFF - Active/Désactive la trace de haut niveau de l'API.
- TRACE TIMESTAMP ON|OFF - Active/Désactive un préfixe timestamp sur chaque ligne tracée.
- TRACE SIZE <size> - Limite la taille du fichier de trace à <size> octets, 8192 octets sont au moins nécessaire.

8.66.23

maxdb_disable_reads_from_master() : maxdb->disable_reads_from_master()

Style procédural

bool **maxdb_disable_reads_from_master** (resource link)

Style orienté objet (méthode)

maxdb void **disable_reads_from_master** ()

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.66.24 maxdb_disable_rpl_parse() : Désactive l'analyseur RPL

bool **maxdb_disable_rpl_parse** (resource link)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.66.25 maxdb_dump_debug_info() : Sauvegarde les informations de débogage dans un log

bool **maxdb_dump_debug_info** (resource link)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.66.26 maxdb_embedded_connect() : Ouvre une connexion sur un serveur MaxDB embarqué

resource **maxdb_embedded_connect** (*string* *dbname*)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.66.27 **maxdb_enable_reads_from_master()** : Active la lecture depuis le maître

bool **maxdb_enable_reads_from_master** (resource *link*)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.66.28 **maxdb_enable_rpl_parse()** : Active l'analyseur RPL

bool **maxdb_enable_rpl_parse** (resource *link*)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.66.29 **maxdb_errno()** : **maxdb->errno()**

Style procédural

int **maxdb_errno** (resource *link*)

Style orienté objet (méthode)

maxdb int *errno*

maxdb_errno retourne le dernier code erreur pour le dernier appel à une fonction MaxDB, qu'elle ait réussie ou échouée, sur la connexion à la base de données représentée par le paramètre *link* . Si aucune erreur n'est survenue, cette fonction retournera zéro.

8.66.30 **maxdb_error()** : Retourne une chaîne représentant la dernière erreur

Style procédural

string **maxdb_error** (resource *link*)

Style orienté objet (méthode)

maxdb string *error*

maxdb_error est identique à la fonction maxdb_errno dans tous les sens, exceptée qu'au lieu de retourner une erreur sous forme d'entier, maxdb_error retourne une chaîne représentant la dernière erreur survenue pour la connexion à la base de données représentée par le paramètre *link* . Si aucune erreur n'est survenue, cette fonction retournera une chaîne vide.

8.66.31 `maxdb_escape_string()` : Alias de `maxdb_real_escape_string`

Cette fonction est un alias de : `maxdb_real_escape_string` .

8.66.32 `maxdb_execute()` : Alias de `maxdb_stmt_execute`

Cette fonction est un alias de : `maxdb_stmt_execute` .

Cet alias est déprécié et n'existe que pour des raisons de compatibilité. Il est recommandé de ne pas utiliser cette fonction car elle peut être supprimée de PHP un jour.

8.66.33 `maxdb_fetch_array()` : `result->fetch_array()`

Style procédural

mixed `maxdb_fetch_array` (resource `result` , int `resulttype`)

Style orienté objet (méthode)

result mixed `fetch_array` (int `resulttype`)

Retourne un tableau qui correspond à la ligne récupérée ou NULL s'il n'y a plus de lignes pour le jeu de résultats représenté par le paramètre `result` .

`maxdb_fetch_array` est une version étendue de la fonction `maxdb_fetch_row` . En plus de stocker les données dans un tableau de résultats aux indices numériques , la fonction `maxdb_fetch_array` peut également stocker les données dans des indices associatifs, en utilisant les noms de champs du résultat en tant que clés.

Note
Les noms des champs retournés par cette fonction sont sensibles à la casse .
Note
Cette fonction définit les champs NULL à la valeur PHP NULL .

Si deux colonnes ou plus du résultats ont les mêmes noms de champs, la dernière colonne aura la priorité et effacera les données précédentes. Pour accéder aux autres colonnes du même nom, la version indexée numériquement de la ligne doit être utilisée.

Le second argument optionnel `resulttype` est une constante indiquant quel type de tableau doit être produit pour les données de la ligne courante. Les valeurs possibles pour ce paramètre sont les constantes `MAXDB_ASSOC` , `MAXDB_ASSOC_UPPER` , `MAXDB_ASSOC_LOWER` , `MAXDB_NUM` , ou `MAXDB_BOTH` . Par défaut, la fonction `maxdb_fetch_array` prendra la valeur `MAXDB_BOTH` , qui est une combinaison de `MAXDB_NUM` et de `MAXDB_ASSOC` , pour ce paramètre.

En utilisant la constante `MAXDB_ASSOC` , cette fonction fonctionne de la même manière que la fonction `maxdb_fetch_assoc` , tandis que `MAXDB_NUM` fonctionne de la même manière que la fonction `maxdb_fetch_row` . L'option `MAXDB_BOTH` créera un seul tableau contenant les deux sortes d'attributs.

En utilisant la constante `MAXDB_ASSOC_UPPER`, le comportement de cette fonction est identique à l'utilisation de `MAXDB_ASSOC` à la seule différence que l'index de la colonne du tableau est le

nom du champs en majuscule.

En utilisant la constante `MAXDB_ASSOC_LOWER`, le comportement de cette fonction est identique à l'utilisation de `MAXDB_ASSOC` à la seule différence que l'index de la colonne du tableau est le nom du champs en minuscule.

8.66.34 `maxdb_fetch_assoc()` : `maxdb->fetch_assoc()`

Style procédural

array `maxdb_fetch_assoc` (resource result)

Style orienté objet (méthode)

result array `fetch_assoc` ()

Retourne un tableau associatif qui correspond à la ligne récupérée ou NULL s'il n'y a plus de lignes de disponibles.

`maxdb_fetch_assoc` est utilisé pour retourner un tableau associatif représentant la prochaine ligne dans le jeu de résultats pour le résultat représenté par le paramètre `result`, où chaque clé du tableau représente le nom d'une des colonnes du jeu de résultats.

Si deux colonnes ou plus du résultats ont les mêmes noms de champs, la dernière colonne aura la priorité. Pour accéder aux autres colonnes du même nom, vous devez soit accéder au résultat avec des indices numériques en utilisant la fonction `maxdb_fetch_row`, soit ajouter des alias aux noms.

Note
Les noms des champs retournés par cette fonction sont sensibles à la casse .
Note
Cette fonction définit les champs NULL à la valeur PHP NULL .

8.66.35 `maxdb_fetch_field_direct()` : `result->fetch_field_direct()`

Style procédural

mixed `maxdb_fetch_field_direct` (resource result , int fieldnr)

Style orienté objet (méthode)

result mixed `fetch_field_direct` (int fieldnr)

`maxdb_fetch_field_direct` retourne une ressource qui contient les informations de définition des champs pour un jeu de résultat spécifique. La valeur du champ doit être dans l'intervalle 0 à `number of fields - 1` .

8.66.36 `maxdb_fetch_field()` : `result->fetch_field()`

Style procédural

mixed `maxdb_fetch_field` (resource result)

8.66.33 `maxdb_fetch_array()` : `result->fetch_array()`

Style orienté objet (méthode)

result mixed **fetch_field** (void)

maxdb_fetch_field retourne la définition d'une colonne d'un jeu de résultats en tant que ressource. Appelez cette fonction répétitivement pour récupérer toutes les informations sur les colonnes dans le jeu de résultats. maxdb_fetch_field retourne FALSE lorsqu'il n'y a plus de champs disponibles.

8.66.37 maxdb_fetch_fields() : result->fetch_fields()

Style procédural

mixed **maxdb_fetch_fields** (resource result)

Style orienté objet (méthode)

result mixed **fetch_fields** (void)

Cette fonction est identique à la fonction maxdb_fetch_field avec la seule différence qu'au lieu de retourner une ressource à la fois pour chaque champ, les colonnes sont retournées en tant que tableau de ressources.

8.66.38 maxdb_fetch_lengths() : result->lengths()

Style procédural

array **maxdb_fetch_lengths** (resource result)

Style orienté objet (méthode)

result array lengths

maxdb_fetch_lengths retourne un tableau contenant la longueur de chaque colonne de la ligne courante du jeu de résultats représenté par le paramètre result . Si cela réussit, un tableau indexé numériquement représentant la longueur de chaque colonne est retourné, ou FALSE si une erreur survient.

8.66.39 maxdb_fetch_object() : result->fetch_object()

Style procédural

object **maxdb_fetch_object** (object result)

Style orienté objet (méthode)

result object **fetch_object** ()

maxdb_fetch_resource retourne la ligne courante du jeu de résultats result dans un objet dont les attributs sont les noms des champs trouvés dans le jeu de résultats. S'il n'y a plus de lignes dans le jeu de résultats courant, NULL est retourné.

8.66.40 **maxdb_fetch_row()** : **result->fetch_row()**

Style procédural

mixed **maxdb_fetch_row** (resource result)

Style orienté objet (méthode)

result mixed **fetch_row** ()

Retourne un tableau qui correspond à la ligne récupérée ou NULL s'il n'y a plus de résultat.

maxdb_fetch_row récupère une ligne de données depuis le jeu de résultats représenté par result et la retourne dans un tableau énuméré, où chaque colonne est stocké dans une entrée du tableau, en commençant à 0 (zéro). Chaque sous-séquence d'appel à la fonction maxdb_fetch_row retourne la ligne suivante du jeu de résultats ou FALSE s'il n'y a plus de lignes de disponibles.

8.66.41 **maxdb_fetch()** : Alias de maxdb_stmt_fetch

Cette fonction est un alias de : maxdb_stmt_fetch .

Cet alias est déprécié et n'existe que pour des raisons de compatibilité. Il est recommandé de ne pas utiliser cette fonction car elle peut être supprimée de PHP un jour.

8.66.42 **maxdb_field_count()** : **maxdb->field_count()**

Style procédural

int **maxdb_field_count** (resource link)

Style orienté objet (méthode)

maxdb int **field_count** ()

Retourne le nombre de colonnes pour la dernière requête sur la connexion représentée par le paramètre link . Cette fonction est utile lors de l'utilisation de la fonction maxdb_store_result pour déterminer si la requête a produit un jeu de résultats non-vide ou non, sans connaître la nature de la requête.

8.66.43 **maxdb_field_seek()** : **result->field_seek()**

Style procédural

bool **maxdb_field_seek** (resource result , int fieldnr)

Style orienté objet (méthode)

result bool **field_seek** (int fieldnr)

Définit le curseur de champ à une position donnée. Le prochain appel à la fonction

[maxdb_fetch_field](#) récupèrera la définition de champ de la colonne associée avec cette position.

Note
Pour placer le curseur au début d'une ligne, passez zéro comme valeur.

8.66.44 maxdb_field_tell() : result->current_field()

Style procédural

int **maxdb_field_tell** (resource result)

Style orienté objet (méthode)

result int current_field

Retourne la position du curseur de champs utilisé par le dernier appel à [maxdb_fetch_field](#) . Cette valeur peut être utilisée en tant qu'argument à la fonction [maxdb_field_seek](#) .

8.66.45 maxdb_free_result() : result->free()

Style procédural

void **maxdb_free_result** (resource result)

Style orienté objet (méthode)

result void **free** ()

[maxdb_free_result](#) libère la mémoire associée avec le résultat représenté par le paramètre result , qui a été allouée par [maxdb_query](#) , [maxdb_store_result](#) ou [maxdb_use_result](#) .

Note
Vous devriez toujours libérer votre résultat avec la fonction maxdb_free_result , lorsque votre ressource de résultat n'est plus utile.

8.66.46 maxdb_get_client_info() : Retourne la version du client MaxDB sous la forme d'une chaîne

string **maxdb_get_client_info** ()

[maxdb_get_client_info](#) est utilisé pour retourner une chaîne représentant la version cliente utilisée dans l'extension MaxDB.

8.66.47 maxdb_get_client_version() : Récupère des informations sur le client MaxDB

int **maxdb_get_client_version** (void)

Retourne le numéro de version du client en tant qu'entier.

8.66.48 **maxdb_get_host_info()** : **maxdb->get_host_info()**

Style procédural

string **maxdb_get_host_info** (resource link)

Style orienté objet (méthode)

maxdb string host_info

maxdb_get_host_info retourne une chaîne décrivant la connexion représentée par le paramètre link .

8.66.49 **maxdb_get_metadata()** : Alias de **maxdb_stmt_result_metadata**

Cette fonction est un alias de : maxdb_stmt_result_metadata .

Cet alias est déprécié et n'existe que pour des raisons de compatibilité. Il est recommandé de ne pas utiliser cette fonction car elle peut être supprimée de PHP un jour.

8.66.50 **maxdb_get_proto_info()** : **maxdb->protocol_version()**

Style procédural

int **maxdb_get_proto_info** (resource link)

Style orienté objet (méthode)

maxdb string protocol_version

Retourne un entier représentant la version du protocole MaxDB utilisé par la connexion représentée par le paramètre link .

8.66.51 **maxdb_get_server_info()** : **maxdb->server_info()**

Style procédural

string **maxdb_get_server_info** (resource link)

Style orienté objet (méthode)

maxdb string server_info

Retourne une chaîne représentant la version du serveur MaxDB sur lequel l'extension MaxDB est connecté (représenté par le paramètre link).

8.66.52 `maxdb_get_server_version()` : Retourne la version du serveur MaxDB sous la forme d'un entier

Style procédural

int `maxdb_get_server_version` (resource link)

Style orienté objet (méthode)

maxdb int server_version

`maxdb_get_server_version` retourne la version du serveur connecté (représenté par le paramètre link) en tant qu'entier.

La forme de ce numéro de version est `main_version * 10000 + minor_version * 100 + sub_version` (i.e. version 7.5.0 est 70500).

8.66.53 `maxdb_info()` : `maxdb->info()`

Style procédural

string `maxdb_info` (resource link)

Style orienté objet (méthode)

maxdb string info

`maxdb_info` retourne une chaîne contenant des informations sur la dernière requête exécutée. La nature de cette chaîne est définie ci-dessous :

Type de requête	Exemple de chaîne résultante
INSERT INTO...SELECT...	Records: 100 Duplicates: 0 Warnings: 0
INSERT INTO...VALUES (...),(...),(...)	Records: 3 Duplicates: 0 Warnings: 0
LOAD DATA INFILE ...	Records: 1 Deleted: 0 Skipped: 0 Warnings: 0
ALTER TABLE ...	Records: 3 Duplicates: 0 Warnings: 0
UPDATE ...	Rows matched: 40 Changed: 40 Warnings: 0
Note	
Les requêtes qui ne correspondent pas à une des catégories ci-dessus ne sont pas supportées. Dans ce cas, <code>maxdb_info</code> retourne une chaîne vide.	

8.66.54 `maxdb_init()` : Initialise MaxDB et retourne une ressource pour l'utiliser avec `maxdb_real_connect`

resource `maxdb_init` (void)

Alloue ou initialise une ressource MaxDB utilisable pour avec les fonctions `maxdb_options` et `maxdb_real_connect` .

Note

Tous les appels de sous-séquence à n'importe quel fonction maxdb (sauf [maxdb_options](#)) échoueront tant que la fonction [maxdb_real_connect](#) ne sera pas appelée.

8.66.55 maxdb_insert_id() : maxdb->insert_id()

Style procédural

mixed **maxdb_insert_id** (resource link)

Style orienté objet (méthode)

maxdb mixed insert_id

[maxdb_insert_id](#) retourne le dernier identifiant généré par une requête sur une table qui a une colonne possédant l'attribut DEFAULT SERIAL. Si la dernière requête n'était pas une requête de type INSERT ou UPDATE ou si la table modifiée n'a pas de colonne possédant un attribut DEFAULT SERIAL, cette fonction retournera zéro.

8.66.56 maxdb_kill() : maxdb->kill()

Style procédural

bool **maxdb_kill** (resource link , int processid)

Style orienté objet (méthode)

maxdb bool **kill** (int processid)

Cette fonction est utilisée pour se déconnecter d'un serveur MaxDB spécifié par le paramètre processid .

8.66.57 maxdb_master_query() : Force l'exécution d'une requête sur le maître dans une configuration maître/esclave

bool **maxdb_master_query** (resource link , string query)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.66.58 maxdb_more_results() : maxdb->more_results()

bool **maxdb_more_results** (resource link)

[maxdb_more_results](#) indique si un ou plusieurs jeux de résultats sont disponibles depuis le dernier appel à la fonction [maxdb_multi_query](#) .

8.66.59 `maxdb_multi_query()` : `maxdb->multi_query()`

Style procédural

bool `maxdb_multi_query` (resource link , string query)

Style orienté objet (méthode)

maxdb bool `multi_query` (string query)

`maxdb_multi_query` fonctionne de la même manière que la fonction `maxdb_query` . Les requêtes multiples ne sont pas encore supportées.

8.66.60 `maxdb_next_result()` : `maxdb->next_result()`

bool `maxdb_next_result` (resource link)

Vu que les requêtes multiples ne sont pas encore implémentées, `maxdb_next_result` retourne toujours FALSE .

8.66.61 `maxdb_num_fields()` : `result->field_count()`

Style procédural

int `maxdb_num_fields` (resource result)

Style orienté objet (méthode)

result int `field_count`

`maxdb_num_fields` retourne le nombre de champs pour un jeu de résultats spécifique.

8.66.62 `maxdb_num_rows()` : Récupère le nombre de lignes dans un résultat

Style procédural

int `maxdb_num_rows` (resource result)

Style orienté objet (méthode)

maxdb int `num_rows`

Retourne le nombre de lignes dans un jeu de résultats.

L'utilisation de la fonction `maxdb_num_rows` dépend si vous utilisez des jeux de résultats bufferisés ou non. Dans le cas où vous utilisez des jeux de résultats non bufferisés, `maxdb_num_rows` ne retournera pas un nombre de lignes correct tant que toutes les lignes du jeu de résultats n'auront pas été récupérées.

8.66.63 maxdb_options() : maxdb->options()

Style procédural

bool **maxdb_options** (resource link , int option , mixed value)

Style orienté objet (méthode)

maxdb bool **options** (int option , mixed value)

maxdb_options peut être utilisé pour définir des options de connexion supplémentaire et affecte le comportement de cette connexion.

Cette fonction peut être appelée plusieurs fois pour définir différentes options.

maxdb_options doit être appelé après maxdb_init et avant maxdb_real_connect .

Le paramètre option est l'option que vous voulez définir, le paramètre value est la valeur de cette option. Pour plus de descriptions sur ces options, lisez <http://dev.mysql.com/doc/maxdb/> . Le paramètre option peut être une des valeurs suivantes :

Nom	Description
MAXDB_COMPNAME	Le composant "name" utilisé pour initialiser l'environnement d'exécution SQLDBC.
MAXDB_APPLICATION	L'application à connecter à la base de données.
MAXDB_APPVERSION	La version de l'application.
MAXDB_SQLMODE	Le mode SQL.
MAXDB_UNICODE	TRUE si la connexion est un client unicode(UCS2), FALSE sinon.
MAXDB_TIMEOUT	Le délai maximal d'inactivité autorisé après lequel la connexion à la base de données sera fermée par le système.
MAXDB_ISOLATIONLEVEL	Spécifie si oui ou non et combien de verrous partagés et de verrous exclusifs sont demandés ou libérés implicitement.
MAXDB_PACKETCOUNT	Le nombre de paquets différents utilisés pour la connexion.
MAXDB_STATEMENTCACHE SIZE	Le nombre de requêtes préparées à être mises en cache pour la connexion pour être réutilisées.
MAXDB_CURSORPREFIX	Le préfixe à utiliser pour les tables résultantes qui sont automatiquement nommées.

8.66.64 maxdb_param_count() : Alias de maxdb_stmt_param_count

Cette fonction est un alias de : maxdb_stmt_param_count .

Cet alias est déprécié et n'existe que pour des raisons de compatibilité. Il est recommandé de ne pas utiliser cette fonction car elle peut être supprimée de PHP un jour.

8.66.65 maxdb_ping() : maxdb->ping()

Style procédural

bool **maxdb_ping** (resource link)

Style orienté objet (méthode)

maxdb bool **ping** (void)

Vérifie si la connexion au serveur est active. Si elle a été interrompue et que l'option globale `maxdb.reconnect` est active, une reconnexion automatique est tentée.

Cette fonction peut être utilisée par les clients qui restent en attente durant de grande période, pour vérifier si la connexion a été interrompue par le serveur et pour se reconnecter si nécessaire.

8.66.66 maxdb_prepare() : maxdb->prepare()

Style procédural

resource **maxdb_prepare** (resource link , string query)

Style orienté objet (méthode)

stmt resource **prepare** (string query)

`maxdb_prepare` prépare la requête SQL pointée par la chaîne de requête et retourne un gestionnaire de requête à utiliser pour les opérations futures sur la requête. La requête doit être composée d'une seule requête SQL.

Note
Vous ne devez pas ajouter de point-virgule ou de <code>\g</code> à la fin de la requête.
Le paramètre <code>query</code> peut inclure un ou plusieurs marqueurs dans la requête SQL en y intégrant des point d'interrogations (<code>?</code>) aux positions voulues.

Note
Les marqueurs sont autorisés uniquement à des positions précises dans les requêtes SQL. Par exemple, ils sont autorisés dans la liste <code>VALUE()</code> d'une requête <code>INSERT</code> (pour spécifier une valeur de colonne pour une ligne), ou dans une comparaison avec une colonne dans une clause <code>WHERE</code> pour spécifier une valeur de comparaison.
Cependant, ils ne sont pas autorisés pour les identifiants (comme les noms de tables ou de colonnes), dans la liste <code>SELECT</code> qui nomme les colonnes à être retournées par les requêtes <code>SELECT</code> , ou pour spécifier les opérands d'un opérateur binaire comme le signe égal (<code>=</code>). La dernière restriction est nécessaire car il est impossible de déterminer le type du paramètre. En général, les paramètres sont légaux uniquement dans les requêtes Data Manipulation Language (DML) et non dans les requêtes Data Definition Language (DDL) .
Les marqueurs doit être liés aux variables en utilisant les fonctions <code>maxdb_stmt_bind_param</code> et/ou <code>maxdb_stmt_bind_result</code> avant d'exécuter la requête ou de récupérer les lignes.

8.66.67 **maxdb_query()** : **maxdb->query()**

Style procédural

mixed **maxdb_query** (resource link , string query , int resultmode)

Style orienté objet (méthode)

maxdb mixed **query** (string query)

maxdb_query est utilisé pour simplifier l'exécution d'une requête sur une base de données représentée par le paramètre link .

8.66.68 **maxdb_real_connect()** : **maxdb->real_connect()**

Style procédural

bool **maxdb_real_connect** (resource link , string hostname , string username , string passwd , string dbname , int port , string socket)

Style orienté objet (méthode)

maxdb bool **real_connect** (string hostname , string username , string passwd , string dbname , int port , string socket)

maxdb_real_connect tente d'ouvrir une connexion sur un serveur MaxDB fonctionnant sur l'hôte hostname .

Cette fonction diffère de maxdb_connect :

- maxdb_real_connect nécessite une ressource valide créée par la fonction maxdb_init .
- Avec la fonction maxdb_options , vous pouvez définir plusieurs options pour la connexion.

8.66.69 **maxdb_real_escape_string()** : **maxdb->real_escape_string()**

Style procédural

string **maxdb_real_escape_string** (resource link , string escapestr)

Style orienté objet (méthode)

maxdb string **real_escape_sring** (string escapestr)

Cette fonction est utilisée pour créer une chaîne SQL légale que vous pouvez utiliser dans une requête SQL. La chaîne escapestr est encodée en une chaîne échappée SQL, en tenant compte du jeu de caractères courant de la connexion.

Les caractères encodés sont ' , " .

8.66.70 `maxdb_real_query()` : `maxdb->real_query()`

Style procédural

bool `maxdb_real_query` (resource link , string query)

Style orienté objet (méthode)

maxdb bool `real_query` (string query)

`maxdb_real_query` est identique au niveau fonctionnalité que la fonction `maxdb_query` .

Note
Pour déterminer si une requête donnée doit retourner un jeu de résultats ou non, référez-vous à la fonction <code>maxdb_field_count</code> .

8.66.71 `maxdb_report()` : Active ou désactive le rapport interne des fonctions

bool `maxdb_report` (int flags)

8.66.72 `maxdb_rollback()` : `maxdb->rollback()`

Style procédural

bool `maxdb_rollback` (resource link)

Style orienté objet (méthode)

maxdb bool `rollback` (void)

Annule la transaction courante pour la base de données spécifiée par le paramètre link .

8.66.73 `maxdb_rpl_parse_enabled()` : Vérifie si l'analyseur RPL est activé

int `maxdb_rpl_parse_enabled` (resource link)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.66.74 `maxdb_rpl_probe()` : Test RPL

bool `maxdb_rpl_probe` (resource link)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.66.75 **maxdb_rpl_query_type() : maxdb->rpl_query_type()**

Style procédural

int **maxdb_rpl_query_type** (resource link)

Style orienté objet (méthode)

maxdb int **rpl_query_type** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.66.76 **maxdb_select_db() : maxdb->select_db()**

bool **maxdb_select_db** (resource link , string dbname)

maxdb_select_db sélectionne la base de données (spécifiée par le paramètre dbname) a être utilisée lors de l'exécution de requête sur la connexion représentée par le paramètre link .

Note

Cette fonction ne doit être utilisée que pour changer de base de données pour une connexion donnée. Vous pouvez sélectionner une base de données par défaut en utilisant le quatrième paramètres dans la fonction <u>maxdb_connect</u> .
--

8.66.77 **maxdb_send_long_data() : Alias de maxdb_stmt_send_long_data**

Cette fonction est un alias de : maxdb_stmt_send_long_data .

Cet alias est déprécié et n'existe que pour des raisons de compatibilité. Il est recommandé de ne pas utiliser cette fonction car elle peut être supprimée de PHP un jour.

8.66.78 **maxdb_send_query() : maxdb->send_query()**

Style procédural

bool **maxdb_send_query** (resource link , string query)

Style orienté objet (méthode)

maxdb bool **send_query** (string query)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.66.79 **maxdb_server_end() : Eteint un serveur embarqué**

void **maxdb_server_end** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.66.80 **maxdb_server_init()** : Initialise un serveur embarqué

bool **maxdb_server_init** (*array* *server* , *array* *groups*)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.66.81 **maxdb_set_opt()** : Alias de **maxdb_options**

Cette fonction est un alias de : [maxdb_options](#) .

8.66.82 **maxdb_sqlstate()** : **maxdb->sqlstate()**

Style procédural

string **maxdb_sqlstate** (resource *link*)

Style orienté objet (méthode)

maxdb string sqlstate

Retourne une chaîne contenant le code erreur SQLSTATE pour la dernière erreur. Le code erreur est composé de cinq caractères. '00000' signifie aucune erreur. Les valeurs sont spécifiées par ANSI SQL et ODBC.

Note

Notez que toutes les erreurs MaxDB ne sont pas encore implémentées dans SQLSTATE. La valeur HY000 (erreur générale) est utilisée pour les erreurs non implémentées.

8.66.83 **maxdb_ssl_set()** : **maxdb->ssl_set()**

Style procédural

bool **maxdb_ssl_set** (resource *link* , string *key* , string *cert* , string *ca* , string *capath* , string *cipher*)

Style orienté objet (méthode)

maxdb bool **ssl_set** (string *key* , string *cert* , string *ca* , string *capath* , string *cipher*)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.66.84 **maxdb_stat()** : **maxdb->stat()**

Style procédural

string **maxdb_stat** (resource link)

Style orienté objet (méthode)

maxdb string **maxdb->stat** ()

maxdb_stat retourne une chaîne contenant plusieurs informations concernant le serveur MaxDB.

8.66.85 **maxdb_stmt_affected_rows()** : **maxdb_stmt->affected_rows()**

Style procédural

int **maxdb_stmt_affected_rows** (resource stmt)

Style orienté objet (méthode)

stmt int **affected_rows**

maxdb_stmt_affected_rows retourne le nombre de lignes affectées par la dernière requête INSERT, UPDATE ou DELETE. Si la dernière requête est invalide ou si le nombre de lignes ne peut être déterminé, cette fonction retournera -1.

8.66.86 **maxdb_stmt_bind_param()** : **stmt->bind_param()**

Style procédural

bool **maxdb_stmt_bind_param** (resource stmt , string types , mixed var1 , *mixed* ...)

Style orienté objet (méthode)

stmt bool **bind_param** (string types , mixed var1 , *mixed* ...)

Style procédural (syntaxe étendue) :

bool **maxdb_stmt_bind_param** (resource stmt , string types , array var)

Style orienté objet (méthode) (syntaxe étendue):

stmt bool **bind_param** (string types , array var)

maxdb_stmt_bind_param est utilisé pour lier les variables pour les marqueurs dans la requête SQL qui a été passée à maxdb_prepare . La chaîne types contient un ou plusieurs caractères qui spécifient le(s) type(s) correspondant aux variables liées.

La syntaxe étendue de maxdb_stmt_bind_param permet de passer les paramètres en tant que tableaux plutôt qu'en tant que listes de variables PHP à la fonction. Si le tableau de variables n'a pas été utilisé avant l'appel à la fonction maxdb_stmt_bind_param , il doit être initialisé en tant que tableau vide. Voir les exemples sur l'utilisation de la fonction maxdb_stmt_bind_param avec la syntaxe étendue.

Les variables pour les requêtes SELECT INTO SQL peuvent également être liées en utilisant la fonction maxdb_stmt_bind_param . Les paramètres pour les procédures de base de données

peuvent être liés en utilisant la fonction [maxdb_stmt_bind_param](#) . Voir les exemples sur l'utilisation de la fonction [maxdb_stmt_bind_param](#) dans ce cas.

Si une variable liée comme variable INTO à une requête SQL a été utilisée avant, le contenu de cette variable sera écrasé par les données de la requête SELECT INTO . Une référence sur cette variable sera invalide après un appel à la fonction [maxdb_stmt_bind_param](#) .

Pour les paramètres INOUT des procédures de base de données, le contenu de la variable liée INOUT sera écrasé par la valeur de sortie de la procédure de la base de données. Une référence sur cette variable sera invalide après un appel à la fonction [maxdb_stmt_bind_param](#) .

Caractères	Description
i	correspond à une variable de type entier
d	correspond à une variable de type &double ;
s	correspond à une variable de type chaîne de caractères
b	correspond à une variable de type blob et sera envoyé dans le paquet

8.66.87 [maxdb_stmt_bind_result\(\)](#) : [stmt->bind_result\(\)](#)

Style procédural

bool [maxdb_stmt_bind_result](#) (resource [stmt](#) , mixed [var1](#) , *mixed* ...)

Style orienté objet (méthode)

[stmt](#) bool [bind_result](#) (mixed [var1](#) , *mixed* ...)

[maxdb_stmt_bind_result](#) est utilisé pour associer (lier) les colonnes du jeu de résultats à des variables. Lorsque [maxdb_stmt_fetch](#) est appelé pour récupérer les données, Le protocole client/serveur MaxDB place les données pour les colonnes liées dans les variables spécifiées [var1](#) ,

Note
<p>Notez que toutes les colonnes doivent être liées avant l'appel à maxdb_stmt_fetch . Les variables liées peuvent être modifiées en leur type PHP correspondant suivant le type des colonnes.</p> <p>Une colonne peut être liée ou reliée à n'importe quel moment, y compris après la récupération d'une partie du jeu de résultats. La nouvelle association prend effet au prochain appel à maxdb_stmt_fetch .</p>

8.66.88 [maxdb_stmt_close_long_data\(\)](#) : [stmt->close_long_data\(\)](#)

Style procédural

bool [maxdb_stmt_close_long_data](#) (resource [stmt](#) , int [param_nr](#))

Style orienté objet (méthode)

[maxdb_stmt](#) bool [maxdb_stmt->close_long_data](#) ()

Cette fonction doit être appelée après une séquence de [maxdb_stmt_send_long_data](#) , qui a été commencée après [maxdb_execute](#) .

param_nr indique quel paramètre à associer avec la fin des données. Les paramètres sont numérotés en commençant à 0.

8.66.89 maxdb_stmt_close() : maxdb_stmt->close()

Style procédural

bool **maxdb_stmt_close** (resource stmt)

Style orienté objet (méthode)

maxdb_stmt bool **maxdb_stmt->close** ()

Ferme une requête préparée. maxdb_stmt_close désalloue également le gestionnaire de requête représenté par stmt . Si la requête courante est en attente ou à des résultats non lus, cette fonction s'interrompt et ainsi, la prochaine requête peut être exécutée.

8.66.90 maxdb_stmt_data_seek() : stmt->data_seek()

Style procédural

bool **maxdb_stmt_data_seek** (resource statement , int offset)

Style orienté objet (méthode)

stmt bool **data_seek** (int offset)

maxdb_stmt_data_seek déplace le pointeur sur une ligne arbitraire offset du jeu de résultat statement . Le paramètre offset doit être compris entre zéro et le nombre total de lignes moins un (0.. maxdb_stmt_num_rows - 1).

8.66.91 maxdb_stmt_errno() : maxdb_stmt->errno()

Style procédural

int **maxdb_stmt_errno** (resource stmt)

Style orienté objet (méthode)

stmt int errno

Pour la requête spécifiée par le paramètre stmt , maxdb_stmt_errno retourne le code erreur pour l'appel le plus récent à une fonction de requête, qu'elle ait réussie ou échouée.

Note
Pour une liste des codes erreurs possibles, lisez la documentation SQLDBC: http://dev.mysql.com/doc/maxdb/ .

8.66.92 maxdb_stmt_error() : maxdb_stmt->error()

Style procédural

string **maxdb_stmt_error** (resource stmt)

Style orienté objet (méthode)

stmt string error

Pour la requête spécifiée par le paramètre `stmt` , `maxdb_stmt_error` retourne un message d'erreur correspondant à la fonction de requête la plus récente, qu'elle ait échouée ou réussie.

8.66.93 maxdb_stmt_execute() : stmt->execute()

Style procédural

bool **maxdb_stmt_execute** (resource stmt)

Style orienté objet (méthode)

stmt bool **execute** ()

`maxdb_stmt_execute` exécute une requête qui a été préparée auparavant en utilisant la fonction `maxdb_prepare` représenté par la ressource `stmt` . Lors de l'exécution, tous les marqueurs existants seront automatiquement remplacés par les données appropriées.

Si la requête est du type UPDATE, DELETE ou INSERT, le nombre total de lignes affectées peut être déterminé en utilisant la fonction `maxdb_stmt_affected_rows` . De même, si la requête rapporte un jeu de résultats, la fonction `maxdb_fetch` est utilisée.

Note

Lors de l'utilisation de la fonction `maxdb_stmt_execute` , la fonction `maxdb_fetch` doit être utilisée pour récupérer les données avant d'effectuer toute autre requête additionnelle.

8.66.94 maxdb_stmt_fetch() : stmt->fetch()

Style procédural

bool **maxdb_stmt_fetch** (resource stmt)

Style orienté objet (méthode)

stmt bool **fetch** ()

`maxdb_stmt_fetch` retourne les données de lignes en utilisant les variables liées par la fonction `maxdb_stmt_bind_result` .

Note

Notez que toutes les colonnes doivent être liées par l'application avant l'appel à la fonction `maxdb_stmt_fetch` .

8.66.95 `maxdb_stmt_free_result()` : `stmt->free_result()`

Style procédural

void **maxdb_stmt_free_result** (resource stmt)

Style orienté objet (méthode)

stmt void **free_result** ()

`maxdb_stmt_free_result` libère la mémoire des résultats associés avec la requête représentée par le paramètre `stmt` , qui a été alloué avec la fonction `maxdb_stmt_store_result` .

8.66.96 `maxdb_stmt_init()` : `maxdb->stmt_init()`

Style procédural

resource **maxdb_stmt_init** (resource link)

Style orienté objet (méthode)

maxdb object **stmt_init** (void)

Alloue et initialise une ressource de requête utilisable pour la fonction `maxdb_stmt_prepare` .

Note
Tous les appels de sous-séquence à n'importe quel fonction <code>maxdb_stmt</code> échoueront tant que la fonction <code>maxdb_stmt_prepare</code> ne sera pas appelée.

8.66.97 `maxdb_stmt_num_rows()` : `stmt->num_rows()`

Style procédural

int **maxdb_stmt_num_rows** (resource stmt)

Style orienté objet (méthode)

stmt int num_rows

Retourne le nombre de lignes dans le jeu de résultats.

8.66.98 `maxdb_stmt_param_count()` : `stmt->param_count()`

Style procédural

int **maxdb_stmt_param_count** (resource stmt)

Style orienté objet (méthode)

stmt int param_count

maxdb_stmt_param_count retourne le nombre de marqueurs présent dans une requête préparée.

8.66.99 maxdb_stmt_prepare() : stmt->prepare()

Style procédural

bool **maxdb_stmt_prepare** (resource stmt , string query)

Style orienté objet (méthode)

stmt mixed **prepare** (string query)

maxdb_stmt_prepare prépare une requête SQL pointée par une chaîne. La ressource de requête doit avoir été allouée par la fonction maxdb_stmt_init . La requête doit être constituée d'une simple requête SQL.

Note

Vous ne devez pas ajouter de point-virgule final ou \g à la requête.

Le paramètre query peut inclure un ou plusieurs marqueurs de paramètres dans la requête SQL en y intégrant des points d'exclamations (?) à une position appropriée.

Note

Les marqueurs sont autorisés uniquement à des positions précises dans les requêtes SQL. Par exemple, ils sont autorisés dans la liste VALUE() d'une requête INSERT (pour spécifier une valeur de colonne pour une ligne), ou dans une comparaison avec une colonne dans une clause WHERE pour spécifier une valeur de comparaison.

Cependant, ils ne sont pas autorisés pour les identifiants (comme les noms de tables ou de colonnes), dans la liste SELECT qui nomme les colonnes à être retournées par les requêtes SELECT, ou pour spécifier les opérants d'un opérateur binaire comme le signe égal (=). La dernière restriction est nécessaire car il est impossible de déterminer le type du paramètre. En général, les paramètres sont légaux uniquement dans les requêtes Data Manipulation Language (DML) et non dans les requêtes Data Defination Language (DDL) .

Les marqueurs doit être liés aux variables en utilisant les fonctions maxdb_stmt_bind_param et/ou maxdb_stmt_bind_result avant d'exécuter la requête ou de récupérer les lignes.

8.66.100 maxdb_stmt_reset() : stmt->reset()

Style procédural

bool **maxdb_stmt_reset** (resource stmt)

Style orienté objet (méthode)

stmt bool **reset** ()

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.66.101 `maxdb_stmt_result_metadata()` : Retourne les méta-données du jeu de résultats pour une requête préparée

Style procédural

resource **maxdb_stmt_result_metadata** (resource stmt)

Style orienté objet (méthode)

stmt resource **result_metadata** ()

Si la requête passée à `maxdb_prepare` est une de celle puisant un jeu de résultats, `maxdb_stmt_result_metadata` retourne la ressource de résultats qui peut être utilisée pour récupérer les méta-données comme, le nombre total de champs et les informations individuelles d'un champs.

Note

Ce pointeur de résultats peut être passé comme argument à n'importe quelle fonction basé sur les champs qui traite les méta-données, comme :

- `maxdb_num_fields`
- `maxdb_fetch_field`
- `maxdb_fetch_field_direct`
- `maxdb_fetch_fields`
- `maxdb_field_count`
- `maxdb_field_seek`
- `maxdb_field_tell`
- `maxdb_free_result`

La structure du jeu de résultats doit être libérée lorsque vous en avez terminé avec, ce qui est obtenu en la passant à la fonction `maxdb_free_result` .

Note

Le jeu de résultat retourné par `maxdb_stmt_result_metadata` contient uniquement des méta-données. Il ne contient aucune ligne de résultats. Les lignes sont obtenues en utilisant le gestionnaire de requête avec la fonction `maxdb_fetch` .

8.66.102 `maxdb_stmt_send_long_data()` : `stmt->send_long_data()`

Style procédural

bool **maxdb_stmt_send_long_data** (resource stmt , int param_nr , string data)

Style orienté objet (méthode)

stmt bool **stmt_send_long_data** (int param_nr , string data)

Permet d'envoyer des données au serveur sous forme de morceaux (ou gros morceaux). Cette fonction peut être appelée plusieurs fois pour envoyer les parties des caractères ou des valeurs de données binaires pour une colonne, qui peut être du type TEXT ou BLOB.

param_nr indique quel paramètre à associer avec les données. Les paramètres sont nummétrés en commençant à 0. data est une chaîne contenant les données à envoyer.

Note

Pour des raisons d'efficacité, cette fonction doit être utilisée après l'appel à la fonction `maxdb_execute`. Dans ce cas, les données ne sont pas stockées côté client. La fin de la séquence doit finir avec un appel à la fonction `maxdb_stmt_close_long_data`.

8.66.103 `maxdb_stmt_sqlstate()` : Retourne une erreur SQLSTATE depuis la requête précédente

string `maxdb_stmt_sqlstate` (resource stmt)

Retourne une chaîne contenant un code erreur SQLSTATE depuis la fonction de requête préparée la plus récente qu'elle ait réussie ou échouée. Le code erreur est composé de cinq caractères. '00000' signifie aucune erreur. La valeur est spécifiée par ANSI SQL et ODBC.

Note

Notez que toutes les erreurs MaxDB ne sont pas encore liées à SQLSTATE. La valeur HY000 (erreur général) est utilisée pour les erreurs non liées.

8.66.104 `maxdb_stmt_store_result()` : `maxdb->store_result()`

Style procédural

bool `maxdb_stmt_store_result` (resource stmt)

Style orienté objet (méthode)

maxdb object `store_result` ()

`maxdb_stmt_store_result` n'a pas de fonctionnalité et ne devrait pas être utilisé pour récupérer les données depuis un serveur MaxDB.

8.66.105 `maxdb_store_result()` : `maxdb->store_result()`

Style procédural

resource `maxdb_store_result` (resource link)

Style orienté objet (méthode)

maxdb object `store_result` ()

Cette fonction n'a aucune fonctionnalité.

8.66.106 `maxdb_thread_id()` : `maxdb->thread_id()`

Style procédural

int `maxdb_thread_id` (resource link)

maxdb_thread_id retourne l'identifiant du thread pour la connexion courante que l'on peut ensuite tuer en utilisant la fonction maxdb_kill . Si la connexion est perdue et que vous vous reconnectez avec la fonction maxdb_ping , l'identifiant du thread changera. De ce fait, vous ne devez récupérer l'identifiant du thread que lorsque vous en avez besoin.

Note

L'identifiant du thread est assigné sur une base "connexion-par-connexion". De ce fait, si la connexion est interrompue puis réétablie, un nouvel identifiant du thread sera assigné.

8.66.107 maxdb_thread_safe() : Indique si le thread safety est donné ou non

Style procédural

bool **maxdb_thread_safe** ()

maxdb_thread_safe indique si la bibliothèque cliente a été compilé en tant que thread safety.

8.66.108 maxdb_use_result() : maxdb->use_result()

Style procédural

resource **maxdb_use_result** (resource link)

Style orienté objet (méthode)

maxdb resource **use_result** ()

maxdb_use_result n'a aucun effet.

8.66.109 maxdb_warning_count() : maxdb->warning_count()

Style procédural

int **maxdb_warning_count** (resource link)

Style orienté objet (méthode)

maxdb int warning_count

maxdb_warning_count retourne le nombre d'avertissements depuis la dernière requête dans la connexion représentée par le paramètre link .

8.67 Chaînes de caractères multi-octets

8.67.1 Introduction

Même si la plupart des langues peuvent être représentées grâce à un jeu de 128 caractères, il y a d'autres langues qui requièrent des jeux de caractères bien plus grands. Des méthodes de caractères multi-octets ont été développées pour résoudre ce type de problème.

Lorsque vous manipulez des chaînes de caractères multi-octets, pour couper, rechercher ou nettoyer une chaîne, vous devez utiliser deux octets consécutifs, qui représentent un seul caractère. Si vous n'y prenez pas garde, vous allez obtenir une chaîne corrompue et invalide, avec une représentation totalement incompréhensible.

mbstring fournit les fonctions spécifiques de manipulations des chaînes multi-octets pour PHP, qui ne supporte que les jeux mono-octets. De plus, mbstring gère la traduction entre les jeux de caractères disponibles.

mbstring est aussi connu pour gérer l'Unicode, comme UTF-8 et UCS-2 ainsi que de nombreux autres jeux mono-octets, même si mbstring a été conçu à la base pour les pages web japonaises.

8.67.1.1 Pré-requis de l'encodage en PHP

Les jeux de caractères suivants sont bien supportés par PHP.

- Jeux de caractères mono-octets :
 - ◆ qui sont compatibles avec la norme ASCII (compatible ISO646) pour les caractères de l'intervalle 00h à 7fh .
- Jeux de caractères multi-octets :
 - ◆ qui sont compatibles avec la norme ASCII (compatible ISO646) pour les caractères de l'intervalle 00h à 7fh .
 - ◆ qui n'utilisent pas les séquences de type ISO2022.
 - ◆ qui n'utilisent pas de valeur dans l'intervalle 00h à 7fh dans les séquences composées, qui représentent un caractère unique.

Les jeux de caractères suivants risquent de ne pas fonctionner en PHP.

Jeux de caractères qui risquent de ne pas fonctionner en PHP
JIS, SJIS, ISO-2022-JP, BIG-5

Même si aucun script PHP écrit avec ces jeux de caractères ne fonctionne, notamment si des chaînes encodées sont utilisées comme identifiants, ou valeurs littérales dans le script, vous pouvez éviter d'utiliser ces jeux en activant le filtre transparent mbstring pour les données d'entrées HTTP.

Note
Il est déconseillé d'utiliser les jeux SJIS, BIG5, CP936, CP949 et GB18030 pour l'encodage interne, à moins que vous ne soyez très familiers avec l'analyseur, le scanneur et le jeu de caractère lui-même.
Note
Si vous avez une base de données connectée à PHP, il est recommandé d'utiliser le même jeu de caractères dans la base et comme jeu interne pour améliorer le confort d'utilisation et les performances.

Si vous utilisez PostgreSQL, le jeu de caractères utilisé dans la base de données et celui de PHP peuvent différer car cette base supporte la traduction automatique de jeu de caractères.

8.67.2 Installation

mbstring est un module PHP. Vous devez activer le module avec le script de configuration configure . Reportez-vous à la section [installation](#) pour plus de détails.

Les options de configurations suivantes sont liées au module mbstring .

- `--enable-mbstring=LANG` : Active les fonctions mbstring . Cette option est nécessaire pour utiliser les fonctions mbstring .

A partir de PHP 4.3.0, l'option `--enable-mbstring` sera activée par défaut, et remplacée par `--with-mbstring[=LANG]` pour supporter le chinois, coréen et russe. Le jeu de caractères japonais est supporté par défaut. Si `--with-mbstring=cn` est utilisé, le chinois simplifié sera supporté. Si `--with-mbstring=tw` est utilisé, le chinois traditionnel sera supporté. Si `--with-mbstring=kr` est utilisé, le coréen sera supporté. Si `--with-mbstring=ru` est utilisé, le russe sera supporté. Si `--enable-mbstring=ja` est utilisé, le japonais sera supporté.

Si `--enable-mbstring=all` est utilisé, tous les langages ci-dessus seront activés.

Note

Le japonais est aussi activé avec `--enable-mbstring` sans autre option pour une compatibilité ascendante.

- `--enable-mbstr-enc-trans` : Active la conversion automatique des données par HTTP, avec le moteur de conversion de mbstring . Si cette option est activée, les données venant du web via HTTP seront converties dans le jeu de caractères `mbstring.internal_encoding` , automatiquement.

Note

Depuis PHP 4.3.0, l'option `--enable-mbstr-enc-trans` sera éliminée, et remplacée par `mbstring.encoding_translation` . La conversion de jeu de caractères d'entrée HTTP sera activée lorsque cette option sera à On (par défaut, cette option vaut Off).

- `--enable-mbregex` : active les fonctions d'expressions rationnelles, compatibles avec les caractères multi-octets.

8.67.3 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable	Historique
<code>mbstring.language</code>	"neutral"	PHP_INI_PERDIR	Disponible depuis PHP 4.3.0
<code>mbstring.detect_order</code>	NULL	PHP_INI_ALL	Disponible depuis PHP 4.0.6
<code>mbstring.http_input</code>	"pass"	PHP_INI_ALL	Disponible depuis PHP 4.0.6
<code>mbstring.http_output</code>	"pass"	PHP_INI_ALL	Disponible depuis PHP 4.0.6
<code>mbstring.internal_encoding</code>	NULL	PHP_INI_ALL	Disponible depuis PHP 4.0.6
<code>mbstring.script_encoding</code>	NULL	PHP_INI_ALL	Disponible depuis PHP 4.3.0
<code>mbstring.substitute_character</code>	NULL	PHP_INI_ALL	Disponible depuis PHP 4.0.6
<code>mbstring.func_overload</code>	"0"	PHP_INI_PERDIR	

			PHP_INI_SYSTEM en PHP <= 4.2.3. Disponible depuis PHP 4.2.0
mbstring.encoding_translation	"0"	PHP_INI_PERDIR	Disponible depuis PHP 4.3.0

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [Directives du php.ini](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

mbstring.language string

Définit le langage utilisé par mbstring. Notez que cette option définit mbstring.internal_encoding mbstring.internal_encoding doit être placé après mbstring.language dans le fichier php.ini

mbstring.encoding_translation boolean

Active la détection et la traduction des données d'entrées HTTP vers le codage interne mbstring.

mbstring.internal_encoding string

Définit l'encodage interne par défaut.

mbstring.http_input string

Définit l'encodage de réception HTTP par défaut.

mbstring.http_output string

Définit l'encodage d'affichage HTTP par défaut.

mbstring.detect_order string

Définit l'ordre de détection des encodages par défaut. Voir aussi [mb_detect_order](#) .

mbstring.substitute_character string

Définit l'encodage de substitution par défaut : il est utilisé pour les caractères invalides.

mbstring.func_overload string

Remplace les fonctions de traitement des chaînes par les fonctions mbstring. Voir les [fonctions de remplacement](#) pour plus d'informations.

En accord avec [HTML 4.01 specification](#) , les navigateurs sont supposés utiliser le même jeu de caractères lorsqu'ils soumettent un formulaire. Mais, tous les navigateurs ne le font pas. Reportez-vous à la fonction [mb_http_input](#) pour détecter les jeux de caractères utilisés par les navigateurs.

En général, les navigateurs sont suffisamment intelligents pour détecter les jeux de caractères dans le HTML. Si charset est activé dans les en-têtes, cela donnera de meilleurs résultats. Changez la valeur de [default_charset](#) avec la valeur appropriée dans le fichier ini.

Configuration de php.ini pour mbstring	
; Langage par défaut	
mbstring.language	= Neutral; (UTF-8) (par défaut)
mbstring.language	= English; Anglais
mbstring.language	= Japanese; Japonais
;; Jeu de caractère interne	
;; Note : assurez-vous que ce jeu fonctionne avec PHP	
mbstring.internal_encoding	= UTF-8
;; Activation de la conversion automatique des entrées HTTP	
mbstring.encoding_translation	= On
;; Jeu de caractères par défaut pour les données d'entrée HTTP	
;; Note : le script ne peut pas changer cette configuration	
mbstring.http_input	= pass ; Aucune conversion.
mbstring.http_input	= auto ; Utilise auto
	; "auto" est remplacé par "ASCII,JIS,UTF-8,EUC-JP,SJIS"
mbstring.http_input	= SJIS ; Utilise SJIS

Manuel PHP 4.3.8 - Version Française

```
mbstring.http_input          = UTF-8,SJIS,EUC-JP ; Spécifie l'ordre
;; Jeu de caractères par défaut pour les données de sortie HTTP
mbstring.http_output        = pass      ; Aucune conversion
mbstring.http_output        = UTF-8    ; Utilise UTF-8
;; Ordre de détection des jeux de caractères
mbstring.detect_order       = auto     ; Utilise la détection automatique
mbstring.detect_order       = ASCII,JIS,UTF-8,SJIS,EUC-JP ; Spécifie l'ordre
;; Déterminer le jeu de caractères de substitution par défaut
mbstring.substitute_character = 12307  ; Spécifie une valeur Unicode
mbstring.substitute_character = none    ; Ne pas afficher de caractères
mbstring.substitute_character = long    ; Exemple complet : U+3000,JIS+7E7E
```

Configuration de php.ini pour les utilisateurs de EUC-JP

```
;; Désactive la bufferisation de sortie
output_buffering          = Off
;; Choisit le jeu de caractères
default_charset           = EUC-JP
;; Le langage par défaut est le japonais
mbstring.language        = Japanese
;; Activation de la traduction automatique des données d'entrée HTTP
mbstring.encoding_translation = On
;; Activation de la conversion automatique
mbstring.http_input      = auto
;; Convertit les sorties en EUC-JP
mbstring.http_output     = EUC-JP
;; Utilise le jeu de caractères interne EUC-JP
mbstring.internal_encoding = EUC-JP
;; Ne pas afficher les caractères invalides
mbstring.substitute_character = none
```

Configuration de php.ini pour les utilisateurs de SJIS

```
;; Active la bufferisation de sortie
output_buffering          = On
;; Utilise le gestionnaire mb_output_handler pour la conversion de sortie
output_handler            = mb_output_handler
;; Choisit le jeu de caractères
default_charset           = Shift_JIS
;; Le langage par défaut est le japonais
mbstring.language        = Japanese
;; Activation de la traduction automatique des données d'entrée HTTP
mbstring.http_input      = auto
;; Convertit en SJIS
mbstring.http_output     = SJIS
;; Utilise le jeu de caractères interne EUC-JP
mbstring.internal_encoding = EUC-JP
;; Ne pas afficher les caractères invalides
mbstring.substitute_character = none
```

8.67.4 Types de ressources

Cette extension ne définit aucune ressource.

8.67.5 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

MB_OVERLOAD_MAIL ([entier](#))
 MB_OVERLOAD_STRING ([entier](#))
 MB_OVERLOAD_REGEX ([entier](#))

8.67.6 Entrées/Sorties HTTP

La conversion automatique des entrées/sorties HTTP peut aussi convertir des données binaires. Les utilisateurs doivent contrôler les conversions, si des données binaires doivent être utilisées via HTTP.

Note

Depuis la version 4.3.2 de PHP, il y a une limitation dans l'utilisation des fonctionnalités mbstring : l'encodage ne sera pas effectué dans les données transmises en méthode POST si l'attribut enctype de la balise form vaut multipart/form-data . Donc, vous devrez convertir les données entrantes vous-même dans ce cas, si nécessaire.

Depuis la version 4.3.3 de PHP, si l'attribut enctype de la balise form vaut multipart/form-data et que la directive du php.ini est positionnée à On, les variables et les noms de fichiers téléchargés en méthode POST seront convertis avec l'encodage interne. Sinon, la conversion ne sera pas faite.

- Entrée HTTP

Il n'y a pas de moyens de contrôler la conversion des caractères HTTP en entrée, depuis un script PHP. Pour désactiver cette conversion, il faut le faire dès le fichier php.ini .

Désactive la conversion HTTP dans le php.ini

```
;; Désactive la conversion HTTP
mbstring.http_input = pass
;; Désactive la conversion HTTP (PHP 4.3.0 ou plus récent)
mbstring.encoding_translation = Off
```

Lorsque vous utilisez PHP comme module Apache, il est possible d'annuler la configuration du php.ini pour chaque Virtual Host dans le fichier httpd.conf ou par dossier avec le fichier .htaccess . Reportez-vous à la section de [configuration](#) ainsi qu'au manuel Apache.

- Sorties HTTP

Il y a plusieurs moyens d'activer la conversion en sortie de script PHP. L'un d'entre eux utilise php.ini , un autre utilise [ob_start](#) avec la fonction [mb_output_handler](#) comme fonction de call-back.

Note

Pour les utilisateurs PHP3-i18n, le système de conversion de mbstring diffère de celui de PHP3-i18n. Le jeu de caractère est converti avec un buffer de sortie.

Exemple de configuration de mbstring dans php.ini

```
;; Active la conversion de sortie pour toutes les pages PHP

;; Active la bufferisation de sortie
output_buffering      = On

;; Choisi mb_output_handler pour effectuer la conversion de sortie
output_handler        = mb_output_handler
```

Exemple de script avec mbstring

```
<?php

// Active la conversion de caractère uniquement pour cette page

// Choisi le jeu de caractères SJIS
mb_http_output('SJIS');

// Commence la bufferisation et spécifie "mb_output_handler"
// comme fonction de callback
ob_start('mb_output_handler');

?>
```

8.67.7 Jeux de caractères supportés

Actuellement, les jeux de caractères suivants sont supportés par mbstring . L'encodage de caractère peut être spécifié par les paramètres encoding dans les fonctions mbstring .

Les jeux de caractères suivants sont supportés par mbstring :

- UCS-4
- UCS-4BE
- UCS-4LE
- UCS-2
- UCS-2BE
- UCS-2LE
- UTF-32
- UTF-32BE
- UTF-32LE
- UTF-16
- UTF-16BE
- UTF-16LE
- UTF-7
- UTF7-IMAP
- UTF-8
- ASCII
- EUC-JP
- SJIS
- eucJP-win
- SJIS-win
- ISO-2022-JP
- JIS
- ISO-8859-1
- ISO-8859-2
- ISO-8859-3
- ISO-8859-4

- ISO-8859-5
- ISO-8859-6
- ISO-8859-7
- ISO-8859-8
- ISO-8859-9
- ISO-8859-10
- ISO-8859-13
- ISO-8859-14
- ISO-8859-15
- byte2be
- byte2le
- byte4be
- byte4le
- BASE64
- HTML-ENTITIES
- 7bit
- 8bit
- EUC-CN
- CP936
- HZ
- EUC-TW
- CP950
- BIG-5
- EUC-KR
- UHC (CP949)
- ISO-2022-KR
- Windows-1251 (CP1251)
- Windows-1252 (CP1252)
- CP866 (IBM866)
- KOI8-R

Les entrées du fichier `php.ini` , qui acceptent des noms de jeux de caractères, acceptent aussi les valeurs " auto " et " pass ". Les fonctions `mbstring` , qui acceptent des noms de jeux de caractères, acceptent aussi la valeur " auto " .

Si " pass " est utilisée, aucune conversion n'est effectuée.

Si " auto " est défini, la liste sera étendue à la liste des encodages définis par NLS . Par exemple, si NLS vaut Japanese , les valeurs seront " ASCII,JIS,UTF-8,EUC-JP,SJIS " .

Voir aussi [mb_detect_order](#) .

8.67.8 Exploitation des chaînes multi-octets en PHP

Comme presque tout PHP est écrit pour des langues qui sont simple-octets, il y a des difficultés à gérer le japonais. Toutes les fonctions PHP de gestion des chaînes de caractères, comme [substr](#) ne supportent pas les chaînes multi-octets.

L'extension Multibyte (multi-octets, alias `mbstring`) dispose de fonctions de chaînes, qui supportent le multi-octet, comme [mb_substr](#) , qui est un remplaçant de [substr](#) .

`mbstring` supporte aussi le remplacement de fonction, pour permettre le support des chaînes multi-octets sans modifier les scripts PHP. En utilisant ce système de remplacement de fonctions,

certaines fonctions PHP seront remplacées par leur équivalent de mbstring. Par exemple `mb_substr` remplacera `substr`. Ce système de remplacement transparent, permet un portage simple et efficace des applications.

`mbstring.func_overload`, dans le `php.ini`, doit être configuré avec une valeur positive : 1 remplace la fonction d'envoi de mail ; 2 active les fonction de chaînes ; 4 active les fonctions d'expression rationnelles. Par exemple, avec la valeur 7, toutes les fonctions possibles sont remplacées. Voici la liste complète des fonctions remplacées, avec leur fonction de remplacement.

Valeur de <code>mbstring.func_overload</code>	Fonction originale	Fonction de remplacement
1	<code>mail</code>	<code>mb_send_mail</code>
2	<code>strlen</code>	<code>mb_strlen</code>
2	<code>strpos</code>	<code>mb_strpos</code>
2	<code> strrpos</code>	<code>mb_strrpos</code>
2	<code>substr</code>	<code>mb_substr</code>
2	<code> strtolower</code>	<code>mb_strtolower</code>
2	<code> strtoupper</code>	<code>mb_strtoupper</code>
2	<code> substr_count</code>	<code>mb_substr_count</code>
4	<code>ereg</code>	<code>mb_ereg</code>
4	<code> eregi</code>	<code>mb_eregi</code>
4	<code>ereg_replace</code>	<code>mb_ereg_replace</code>
4	<code> eregi_replace</code>	<code>mb_eregi_replace</code>
4	<code>split</code>	<code>mb_split</code>
Note		
Il n'est pas recommandé d'utiliser les options des fonctions d'overloading dans un contexte de per-directory, car il n'est pas confirmé encore qu'elles sont suffisamment stables dans un environnement de production et peuvent conduire à un résultat incohérent.		

8.67.9 Cas des caractères japonais

La plupart des caractères japonais demandent plus d'un octet pour être représentés. De plus, plusieurs jeux de caractères japonais existent : il y a notamment EUC-JP, Shift_JIS et ISO-2022-JP. Unicode devient de plus en plus populaire, et UTF-8 aussi. Pour développer des applications Web en environnement japonais, il faut savoir que les encodages ci-dessus dépendent de l'application qu'on en fait : entrée/sortie HTTP, bases de données ou courrier électronique.

- La taille nécessaire à un caractère peut aller jusqu'à 4 octets.
- Un caractère multi-octets occupe généralement deux octets, à comparer avec les caractères simple-octet traditionnellement utilisés. Les caractères les plus gros sont appelés "zen-kaku" (i.e. grande largeur) et les plus petits sont appelés "han-kaku" (i.e. demi-largeur). Les caractères "zen-kaku" sont généralement de taille constante.
- Certains encodages de caractères définissent des séquences de début/fin pour les sections multi-octets (00h to 7fh).
- ISO-2022-JP doit être utilisé pour les protocoles SMTP/NNTP, et les en-têtes ainsi que les entités devraient être réencodés en accord avec la RFC correspondante. Bien que cela ne soit pas requis, ça reste une bonne idée car beaucoup de user-agent (agents utilisateurs) populaires ne peuvent pas reconnaître d'autre méthode d'encodage.
- Les pages Web créées pour les téléphones portables comme [i-mode](#), [Vodafone live!](#), ou [EZweb](#) sont supposées utiliser l'encodage Shift_JIS.

8.67.10 Références

Les jeux de caractères multi-octets et leurs techniques sont très complexes. Il n'est pas possible de couvrir tous les aspects en détails ici. Reportez-vous aux URL suivantes pour d'autres ressources complémentaires :

- Unicode/UTF/UCS/etc

<http://www.unicode.org/>

- Japonais/coréen/Chinois

<http://examples.oreilly.com/cjkvinfo/doc/cjk.inf>

8.67.11 Jeux de caractères supportés

Jeux de caractères supportés Nom dans le registre IANA Jeux de caractères Description Notes
 ISO-10646-UCS-4 ISO 10646 Le jeu de caractères universel (Universal Character Set), avec 31 bits par caractère, au standard UCS-4 par ISO/IEC 10646 . Il est synchronisé avec la dernière version d'Unicode. Si ce nom est utilisé dans l'outil de conversion, le convertisseur essaie de reconnaître le texte à partir du dernier BOM (byte order mark), pour connaître l'ordre des bits.
 ISO-10646-UCS-4 UCS-4 Voir ci-dessus. Contrairement à UCS-4 , les chaînes sont supposées être au format big endian.
 ISO-10646-UCS-4 UCS-4 Voir ci-dessus. Contrairement à UCS-2 , les chaînes sont supposées être au format little endian.
 ISO-10646-UCS-2 UCS-2 Le jeu de caractères universel (Universal Character Set), avec 16 bits par caractère, au standard UCS-2 par ISO/IEC 10646 . Il est synchronisé avec la dernière version d'Unicode. Si ce nom est utilisé dans l'outil de conversion, le convertisseur essaie de reconnaître le texte à partir du dernier BOM (byte order mark), pour connaître l'ordre des bits.
 ISO-10646-UCS-2 UCS-2 Voir ci-dessus. Contrairement à UCS-4 , les chaînes sont supposées être au format big endian.
 UTF-32 Unicode Format de transformation d'Unicode, de 32 bits, dont les cartes correspondent au jeu standard Unicode. Ce jeu n'est pas identique à UCS-4 car les caractères Unicode étaient limités à des valeurs de 21 bits. Si ce nom est utilisé dans l'outil de conversion, le convertisseur essaie de reconnaître le texte à partir du dernier BOM (byte order mark), pour connaître l'ordre des bits.
 UTF-32BE Unicode Voir ci-dessus. Contrairement à UTF-32 , les chaînes sont supposées être au format big endian.
 UTF-32LE Unicode Voir ci-dessus. Contrairement à UTF-32 , les chaînes sont supposées être au format little endian.
 UTF-16 Unicode Format de transformation d'Unicode sur 16 bits. Il faut noter que UTF-16 n'est plus identique à UCS-2 car un mécanisme a été introduit en Unicode 2.0 et UTF-16 fait maintenant référence à un codage de 21 bits. Si ce nom est utilisé dans l'outil de conversion, le convertisseur essaie de reconnaître le texte à partir du dernier BOM (byte order mark), pour connaître l'ordre des bits.
 UTF-16BE Unicode Voir ci-dessus. Contrairement à UTF-16 , les chaînes sont supposées être au format big endian.
 UTF-16LE Unicode Voir ci-dessus. Contrairement à UTF-16 , les chaînes sont supposées être au format little endian.
 UTF-8 Unicode / UCS Format de transformation Unicode de 8 bits. none UTF-7 Unicode Un format compatible avec le courrier électronique d'Unicode, spécifié dans [RFC2152](#) . none aucun Unicode Une variante d' UTF-7 qui est spécialement utilisée dans le [protocole IMAP](#) . none US-ASCII (recommandé) / iso-ir-6 / ANSI_X3.4-1986 / ISO_646.irv:1991 / ASCII / ISO646-US / us / IBM367 / CP367 / csASCII ASCII / ISO 646 ASCII, American Standard Code for Information Interchange est un format classique de 7 bits. Il est aussi normalisé internationalement, sous le nom ISO 646 . (none) EUC-JP (recommandé) / Extended_UNIX_Code_Packed_Format_for_Japanese / csEUCPkFmtJapanese Composé de US-ASCII / JIS X0201:1997 (hankaku kana) / JIS X0208:1990 / JIS X0212:1990 Comme vous le voyez, le nom est dérivé de l'abréviation de Extended UNIX Code Packed Format for Japanese , ce jeu est essentiellement utilisé sur les plates-formes Unix. Le jeu original, Extended UNIX Code , est conçu sur la base de ISO 2022 . Le jeu identifié par EUC-JP est différent de IBM932 / CP932 , qui est utilisé par OS/2® et Microsoft® Windows®. Pour échanger des

informations avec ces plates-formes, utilisez EUCJP-WIN . Shift_JIS (recommandé) / MS_Kanji / csShift_JIS Composé de JIS X0201:1997 / JIS X0208:1997 Shift_JIS a été développé au début des années 80, et, au même moment, les premiers traitements de textes étaient mis sur le marché. Il a été fait pour conserver la compatibilité avec le jeu JIS X 0201:1976 . Selon la définition de l'IANA, le jeu de caractères Shift_JIS est légèrement différent de IBM932 / CP932 . Cependant, les noms "SJIS" et "Shift_JIS" sont souvent utilisés à tort, pour ces jeux. Pour CP932 , utilisez SJIS-WIN . (none) Composé de JIS X0201:1997 / JIS X0208:1997 / IBM extensions / NEC extensions Même si ce "jeu de caractères" utilise le même jeu que EUC-JP , il est en fait différent. Il a juste quelques caractères de différence. none Windows-31J / csWindows31J Composé de JIS X0201:1997 / JIS X0208:1997 / IBM extensions / NEC extensions Même si ce "jeu de caractères" utilise le même jeu que Shift_JIS , il est en fait différent. Il a juste quelques caractères de différence. (none) ISO-2022-JP (recommandé) / csISO2022JP US-ASCII / JIS X0201:1976 / JIS X0208:1978 / JIS X0208:1983 [RFC1468](#) aucun JIS ISO-8859-1 ISO-8859-2 ISO-8859-3 ISO-8859-4 ISO-8859-5 ISO-8859-6 ISO-8859-7 ISO-8859-8 ISO-8859-9 ISO-8859-10 ISO-8859-13 ISO-8859-14 ISO-8859-15 byte2be byte2le byte4be byte4le BASE64 HTML-ENTITIES 7bit 8bit EUC-CN CP936 HZ EUC-TW CP950 BIG-5 EUC-KR UHC (CP949) ISO-2022-KR Windows-1251 (CP1251) Windows-1252 (CP1252) CP866 (IBM866) KOI8-R

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Entrées/Sorties HTTP](#)
- [Jeux de caractères supportés](#)
- [Exploitation des chaînes multi-octets en PHP](#)
- [Cas des caractères japonais](#)
- [Références](#)
- [Jeux de caractères supportés](#)
- [mb_convert_case](#)
- [mb_convert_encoding](#)
- [mb_convert_kana](#)
- [mb_convert_variables](#)
- [mb_decode_mimeheader](#)
- [mb_decode_numericentity](#)
- [mb_detect_encoding](#)
- [mb_detect_order](#)
- [mb_encode_mimeheader](#)
- [mb_encode_numericentity](#)
- [mb_ereg_match](#)
- [mb_ereg_replace](#)
- [mb_ereg_search_getpos](#)
- [mb_ereg_search_getregs](#)
- [mb_ereg_search_init](#)
- [mb_ereg_search_pos](#)
- [mb_ereg_search_regs](#)
- [mb_ereg_search_setpos](#)
- [mb_ereg_search](#)
- [mb_ereg](#)
- [mb_eregi_replace](#)
- [mb_eregi](#)
- [mb_get_info](#)
- [mb_http_input](#)
- [mb_http_output](#)

- [mb_internal_encoding](#)
- [mb_language](#)
- [mb_list_encodings](#)
- [mb_output_handler](#)
- [mb_parse_str](#)
- [mb_preferred_mime_name](#)
- [mb_regex_encoding](#)
- [mb_regex_set_options](#)
- [mb_send_mail](#)
- [mb_split](#)
- [mb_strcut](#)
- [mb_strimwidth](#)
- [mb_strlen](#)
- [mb_strpos](#)
- [mb_strrpos](#)
- [mb_strtolower](#)
- [mb_strtoupper](#)
- [mb_strwidth](#)
- [mb_substitute_character](#)
- [mb_substr_count](#)
- [mb_substr](#)

8.67.13 [mb_convert_encoding\(\)](#) : Conversion d'encodage

string **mb_convert_encoding** (string *str* , string *to_encoding* , *mixed from_encoding*)

[mb_convert_encoding](#) convertit la chaîne *str* depuis l'encodage *from_encoding* vers l'encodage *to_encoding* .

str : chaîne de caractères à convertir.

from_encoding est l'encodage de la chaîne *str* à l'origine. Il sera détecté parmi plusieurs encodages fournis sous forme d'un tableau, ou d'une liste d'encodages séparés par des virgules.

Exemple avec [mb_convert_encoding](#)

```
<?php
/* Convertit l'encodage interne vers SJIS */
$str = mb_convert_encoding($str, "SJIS");
/* Convertit EUC-JP en UTF-7 */
$str = mb_convert_encoding($str, "UTF-7", "EUC-JP");
/* Détecte automatiquement un encodage entre JIS, eucjp-win ou sjis-win,
   Puis convertit en UCS-2LE */
$str = mb_convert_encoding($str, "UCS-2LE", "JIS, eucjp-win, sjis-win");
/* "auto" signifie "ASCII, JIS, UTF-8, EUC-JP, SJIS" */
$str = mb_convert_encoding($str, "EUC-JP", "auto");
?>
```

Voir aussi [mb_detect_order](#) .

8.67.14 [mb_convert_kana\(\)](#) : Convertit entre les différents "kana"

string **mb_convert_kana** (string *str* , string *option* , string *encoding*)

`mb_convert_kana` effectue une conversion "han-kaku" - "zen-kaku" sur la chaîne `str`. Elle retourne la chaîne convertie. Cette fonction n'est utile que pour le japonais.

`option` est l'option de conversion. La valeur par défaut est "KV".

`encoding` est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Spécifiez les conversions en combinant les valeurs suivantes. La valeur par défaut est KV.

Option	Valeur
r	Convertit l'alphabet "zen-kaku" en "han-kaku".
R	Convertit l'alphabet "han-kaku" en "zen-kaku".
n	Convertit les nombres "zen-kaku" en "han-kaku".
N	Convertit les nombres "han-kaku" en "zen-kaku".
a	Convertit les nombres et alphabet "zen-kaku" en "han-kaku".
A	Convertit les nombres et alphabet "zen-kaku" en "han-kaku". (Les caractères inclus dans les options "a", "A" sont U+0021 - U+007E en excluant U+0022, U+0027, U+005C, U+007E)
s	Convertit "zen-kaku" en "han-kaku" (U+3000 -> U+0020)
S	Convertit "han-kaku" en "zen-kaku" (U+0020 -> U+3000)
k	Convertit "zen-kaku kata-kana" en "han-kaku kata-kana"
K	Convertit "han-kaku kata-kana" en "zen-kaku kata-kana".
h	Convertit "zen-kaku hira-gana" en "han-kaku kata-kana".
H	Convertit "han-kaku kata-kana" en "zen-kaku hira-gana".
c	Convertit "zen-kaku kata-kana" en "zen-kaku hira-gana".
C	Convertit "zen-kaku hira-gana" en "zen-kaku kata-kana".
V	Supprime les notations vocales, et les convertit en caractères. A utiliser avec "K","H".

Exemple avec `mb_convert_kana`

```
<?php
/* Convertit tous les "kana" en "zen-kaku" "kata-kana" */
$str = mb_convert_kana($str, "KVC");
/* Convertit "han-kaku" "kata-kana" en "zen-kaku" "kata-kana"
   et "zen-kaku" alpha-numeric en "han-kaku" */
$str = mb_convert_kana($str, "KVa");
?>
```

8.67.15 `mb_convert_variables()` : Convertit l'encodage de variables

string `mb_convert_variables` (string `to_encoding` , mixed `from_encoding` , mixed `vars` , *mixed* ...)

`mb_convert_variables` convertit l'encodage des variables `vars` depuis l'encodage `from_encoding` vers l'encodage `to_encoding`, puis retourne le nom de l'encodage détecté, en cas de succès, ou FALSE en cas d'échec.

`mb_convert_variables` place les chaînes dans un tableau ou un objet pour détecter l'encodage, mais la détection a tendance à échouer pour les chaînes de caractères de petites tailles. De ce fait, il est impossible de mixer les encodages dans un tableau ou un objet "simple".

from-encoding est une liste d'encodages possibles pour les variables vars , fourni sous forme d'un tableau ou d'une liste d'encodages, séparés par des virgules. Si from_encoding est omis, les encodages fournis dans [mb_detect_order](#) sont utilisés.

vars est une référence sur une variable à convertir. Les chaînes, tableaux et objets sont aussi supportés. [mb_convert_variables](#) prend tous ces paramètres avec le même encodage.

Exemple avec [mb_convert_variables](#)

```
<?php
/* Convertit les variables $post1, $post2 en encodage interne */
$interenc = mb_internal_encoding();
$inputenc = mb_convert_variables($interenc, "ASCII,UTF-8,SJIS-win", $post1, $post2);
?>
```

8.67.16 [mb_decode_mimeheader\(\)](#) : Décode un en-tête MIME

string [mb_decode_mimeheader](#) (string str)

[mb_decode_mimeheader](#) décode la chaîne encodée str dans l'en-tête MIME.

[mb_decode_mimeheader](#) retourne la chaîne décodée, encodée au format interne.

Voir aussi [mb_encode_mimeheader](#) .

8.67.17 [mb_decode_numericentity\(\)](#) : Décode les entités HTML en caractères

string [mb_decode_numericentity](#) (string str , array convmap , string encoding)

[mb_decode_numericentity](#) la chaîne d'entités HTML str en chaîne, et retourne cette chaîne.

convmap est un tableau qui spécifie les codes à convertir.

encoding est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Exemple avec le paramètre convmap

```
<?php
$convmmap = array (
    int start_code1, int end_code1, int offset1, int mask1,
    int start_code2, int end_code2, int offset2, int mask2,
    .....
    int start_codeN, int end_codeN, int offsetN, int maskN );
// Spécifie les valeurs Unicode de début (start_codeN) et fin (end_codeN)
// Ajoutez offsetN à la valeur, et faites un ET bit-à-bit avec maskN, puis
// il convertit la valeur obtenue en entité numérique
?>
```

Voir aussi [mb_encode_numericentity](#) .

8.67.18 [mb_detect_encoding\(\)](#) : Détecte un encodage

string [mb_detect_encoding](#) (string str , mixed encoding_list , bool strict)

`mb_detect_encoding` détecte l'encodage utilisé par la chaîne `str`. `mb_detect_encoding` retourne le nom de l'encodage détecté.

`encoding_list` est une liste d'encodages, sous forme de tableau, ou bien de chaîne, les valeurs étant séparées par des virgules.

Si `encoding_list` est omis, l'ordre spécifié par `mb_detect_order` est utilisé.

Exemple avec `mb_detect_encoding`

```
<?php
/* Détecte l'encodage avec les valeurs par défaut */
echo mb_detect_encoding($str);

/* "auto" signifie "ASCII,JIS,UTF-8,EUC-JP,SJIS" */
echo mb_detect_encoding($str, "auto");

/* Spécifie une liste d'encodages possibles avec une liste à virgules */
echo mb_detect_encoding($str, "JIS, eucjp-win, sjis-win");

/* Spécifie une liste d'encodages possibles avec un tableau */
$array[] = "ASCII";
$array[] = "JIS";
$array[] = "EUC-JP";
echo mb_detect_encoding($str, $array);
?>
```

Voir aussi `mb_detect_order`.

8.67.19 `mb_detect_order()` : Lit/modifie l'ordre de détection des encodages

mixed `mb_detect_order` (*mixed encoding_list*)

`mb_detect_order` remplace l'ordre de détection des encodages courant par `encoding_list`.

`mb_detect_order` retourne TRUE en cas de succès, et FALSE en cas d'erreur.

`encoding_list` est un tableau, ou une liste d'encodages séparés par une virgule. La valeur "auto" est automatiquement remplacée par "ASCII, JIS, UTF-8, EUC-JP, SJIS".

Si `encoding_list` est omis, `mb_detect_order` retourne l'ordre de détection courant des encodages.

Ce paramétrage affecte les fonctions `mb_detect_encoding` et `mb_send_mail`.

Note

Actuellement, `mbstring` supporte les filtres de détections ci-dessous. Si une séquence d'octets est invalide pour l'un des filtres suivants, la détection échouera.

UTF-8 , UTF-7 , ASCII , EUC-JP , SJIS , eucJP-win , SJIS-win , JIS et ISO-2022-JP

Pour ISO-8859-* , `mbstring` détecte toujours ISO-8859-* .

Pour UTF-16 , UTF-32 , UCS2 et UCS4 la détection échouera toujours.

Exemple d'ordre de détection inutile

```
; Always detect as ISO-8859-1
```

```

detect_order = ISO-8859-1, UTF-8

; Always detect as UTF-8, since ASCII/UTF-7 values are
; valid for UTF-8
detect_order = UTF-8, ASCII, UTF-7

```

Exemple avec `mb_detect_order`

```

<?php
/* Remplace l'ordre de détection par une liste énumérée */
mb_detect_order("eucjp-win,sjis-win,UTF-8");

/* Remplace l'ordre de détection par un tableau */
$array[] = "ASCII";
$array[] = "JIS";
$array[] = "EUC-JP";
mb_detect_order($array);

/* Affiche l'ordre de détection courant */
echo implode(", ", mb_detect_order());
?>

```

Voir aussi [mb_internal_encoding](#) , [mb_http_input](#) , [mb_http_output](#) et [mb_send_mail](#)

8.67.20 `mb_encode_mimeheader()` : Encode une chaîne pour un en-tête MIME

`string mb_encode_mimeheader (string str , string charset , string transfer_encoding , string linefeed)`

`mb_encode_mimeheader` encode la chaîne `str` en un en-tête MIME, et retourne la chaîne convertie représentée en ASCII.

`charset` est le nom de l'encodage utilisé par la chaîne `str` . La valeur par défaut est déterminée par les paramètres courants du NLS (`mbstring.language`) .

`transfer_encoding` est l'encodage de transfert. Il peut être "B" (Base64) ou "Q" (Quoted-Printable). Par défaut, c'est "B" .

`linefeed` spécifie les fins de lignes (EOF : end-of-line) utilisé par `mb_encode_mime_header` pour formater la chaîne (une [RFC](#) définit la longueur d'une chaîne à partir de laquelle on doit ajouter une fin de ligne. La longueur courante est 74 caractères). La valeur par défaut est "\r\n" (CRLF).

Exemple avec `mb_convert_kana`

```

<?php
$name = ""; // kanji
$mbox = "kru";
$doma = "gtinn.mon";
$addr = mb_encode_mimeheader($name, "UTF-7", "Q") . "<" . $mbox . "@" . $doma . ">";
echo $addr;
?>

```

Note

Cette fonction n'est pas faite pour couper les lignes en milieu de mots. Ce comportement peut ajouter des espaces non désirées dans un mot de la chaîne originale.

Voir aussi [mb_decode_mimeheader](#) .

8.67.21 mb_encode_numericentity() : Encode des entités HTML

string **mb_encode_numericentity** (string str , array convmap , string encoding)

mb_encode_numericentity convertit la chaîne str depuis l'encodage interne en codes numériques HTML, puis retourne cette chaîne.

convmap est un tableau qui spécifie les codes à convertir.

encoding est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Exemple de paramètre convmap

```
<?php
$convmap = array (
    int start_code1, int end_code1, int offset1, int mask1,
    int start_code2, int end_code2, int offset2, int mask2,
    .....
    int start_codeN, int end_codeN, int offsetN, int maskN );
// Spécifie les valeurs Unicode de début (start_codeN) et fin (end_codeN)
// Ajoutez offsetN à la valeur, et faites un ET bit-à-bit avec maskN, puis
// il convertit la valeur obtenue en entité numérique
?>
```

Exemple avec mb_encode_numericentity

```
<?php
/* Convertit du ISO-8859-1 en entités HTML */
$convmap = array(0x80, 0xff, 0, 0xff);
$str = mb_encode_numericentity($str, $convmap, "ISO-8859-1");

/* Convertit du code SJIS-win (uniquement le bloc 95-104) en entités numériques */
$convmap = array(
    0xe000, 0xe03e, 0x1040, 0xffff,
    0xe03f, 0xe0bb, 0x1041, 0xffff,
    0xe0bc, 0xe0fa, 0x1084, 0xffff,
    0xe0fb, 0xe177, 0x1085, 0xffff,
    0xe178, 0xe1b6, 0x10c8, 0xffff,
    0xe1b7, 0xe233, 0x10c9, 0xffff,
    0xe234, 0xe272, 0x110c, 0xffff,
    0xe273, 0xe2ef, 0x110d, 0xffff,
    0xe2f0, 0xe32e, 0x1150, 0xffff,
    0xe32f, 0xe3ab, 0x1151, 0xffff );
$str = mb_encode_numericentity($str, $convmap, "sjis-win");
?>
```

Voir aussi mb_decode_numericentity .

8.67.22 mb_ereg_match() : Expression rationnelle POSIX pour les chaînes multi-octets

bool **mb_ereg_match** (string pattern , string string , string option)

mb_ereg_match retourne TRUE si string vérifie l'expression rationnelle pattern , FALSE sinon.

Le jeu de caractère interne ou bien le jeu spécifié par la fonction mb_regex_encoding sera utilisé comme jeu de caractères.

Voir aussi [mb_regex_encoding](#) et [mb_ereg](#) .

8.67.23 **mb_ereg_replace()** : Remplace des segments de chaînes, avec le support des expressions rationnelles mutli-octets

string **mb_ereg_replace** (string pattern , string replacement , string string , *string option*)

[mb_ereg_replace](#) analyse la chaîne string pour trouver des segments de chaînes correspondant au masque pattern , puis, elle les remplace avec le texte de remplacement replacement et retourne le résultat ou bien FALSE en cas d'erreur. Des caractères multi-octets peuvent être utilisés dans le masque pattern .

Des options de recherches peuvent être configurées avec le paramètre option . Si i est spécifié, la casse sera ignorée. Si x est spécifié, les espaces blancs seront ignorés. Si m est spécifié, la recherche se fera en mode multi-ligne, et les nouvelles lignes seront incluses dans le joker . . Si p est spécifié, la recherche se fera en mode POSIX, les nouvelles lignes seront considérées comme des caractères normaux. Si e est spécifiée, replacement sera évaluée comme une expression PHP.

Le jeu de caractères interne, ou le jeu de caractères est spécifié par [mb_regex_encoding](#) .

Voir aussi [mb_regex_encoding](#) et [mb_eregi_replace](#) .

8.67.24 **mb_ereg_search_getpos()** : Retourne l'offset du début du prochain segment repéré par une expression rationnelle

int **mb_ereg_search_getpos** (void)

[mb_ereg_search_getpos](#) retourne l'offset du prochain segment de code qui correspond au masque utilisé dans [mb_ereg_search](#) , [mb_ereg_search_pos](#) et [mb_ereg_search_regs](#) . L'offset est le nombre de caractères depuis le début de la chaîne.

Le jeu de caractères interne, ou le jeu de caractères est spécifié par [mb_regex_encoding](#) .

Voir aussi [mb_regex_encoding](#) et [mb_ereg_search_setpos](#) .

8.67.25 **mb_ereg_search_getregs()** : Lit le dernier segment de chaîne multi-octets qui correspond au masque

array **mb_ereg_search_getregs** (void)

[mb_ereg_search_getregs](#) retourne un tableau incluant toutes les sous-chaînes qui ont été trouvées par [mb_ereg_search](#) , [mb_ereg_search_pos](#) et [mb_ereg_search_regs](#) . S'il y a une solution qui a été trouvée, le premier élément sera la sous-chaîne trouvée, le second représentera la première parenthèse capturante, le troisième représentera la deuxième parenthèse capturante, etc. Cette fonction retourne FALSE en cas d'erreur.

Le jeu de caractères interne, ou le jeu de caractères est spécifié par [mb_regex_encoding](#) .

Voir aussi [mb_regex_encoding](#) et [mb_ereg_search_init](#) .

8.67.26 `mb_ereg_search_init()` : Configure les chaînes et les expressions rationnelles pour le support des caractères multi-octets

`bool mb_ereg_search_init (string string , string pattern , string option)`

`mb_ereg_search_init` configure `string` et `pattern` pour supporter les expressions rationnelles multi-octets. Ces valeurs sont utilisées par `mb_ereg_search` , `mb_ereg_search_pos` et `mb_ereg_search_regs` . Cette fonction retourne `TRUE` en cas de succès, et `FALSE` en cas d'erreur.

Le jeu de caractères interne, ou le jeu de caractères est spécifié par `mb_regex_encoding` .

Voir aussi `mb_regex_encoding` et `mb_ereg_search_regs` .

8.67.27 `mb_ereg_search_pos()` : Retourne la position et la longueur du segment de chaîne qui vérifie le masque de l'expression rationnelle

`array mb_ereg_search_pos (string pattern , string option)`

`mb_ereg_search_pos` retourne un tableau contenant la position et la taille du segment repéré grâce à l'expression rationnelle. Le premier élément du tableau retourné doit être le début du segment, et le second est la taille du segment, exprimée en octets. Cette fonction retourne `FALSE` en cas d'erreur.

La chaîne à utiliser est spécifiée par `mb_ereg_search_init` . Si elle n'est pas spécifiée, la précédente sera utilisée.

Le jeu de caractères interne, ou le jeu de caractères spécifié par `mb_regex_encoding` sera utilisé.

Voir aussi `mb_regex_encoding` et `mb_ereg_search_init` .

8.67.28 `mb_ereg_search_regs()` : Retourne le segment de chaîne trouvé par une expression rationnelle multi-octets

`array mb_ereg_search_regs (string pattern , string option)`

`mb_ereg_search_regs` exécute l'expression rationnelle `pattern` , et, si un segment de chaîne correspond, elle le retourne dans un tableau, dont le premier élément est le segment de chaîne trouvé, le deuxième le contenu de la première parenthèse capturante, le troisième le contenu de la seconde parenthèse capturante, etc. La fonction retourne `FALSE` en cas d'erreur.

Le jeu de caractères interne ou bien le jeu de caractères spécifié par `mb_regex_encoding` sera utilisé.

Voir aussi `mb_regex_encoding` et `mb_ereg_search_init` .

8.67.29 `mb_ereg_search_setpos()` : Choisit le point de départ de la recherche par expression rationnelle

`bool mb_ereg_search_setpos (int position)`

mb_ereg_search_setpos choisit le point de départ de la recherche que va effectuer la fonction mb_ereg_search .

Le jeu de caractères internet ou bien le jeu de caractères spécifié par mb_regex_encoding sera utilisé.

Voir aussi mb_regex_encoding et mb_ereg_search_init .

8.67.30 mb_ereg_search() : Recherche par expression rationnelle multi-octets

bool **mb_ereg_search** (*string pattern* , *string option*)

mb_ereg_search retourne TRUE si la chaîne multi-octets correspond au masque d'expression rationnelle, ou bien FALSE sinon. La chaîne à étudier a été configurée avec la fonction mb_ereg_search_init . Si le masque pattern n'est pas spécifié, le précédent sera utilisé.

Le jeu de caractères interne, ou le jeu de caractères est spécifié par mb_regex_encoding .

Voir aussi mb_regex_encoding et mb_ereg_search_init .

8.67.31 mb_ereg() : Recherche par expression rationnelle avec support des caractères multi-octets

int **mb_ereg** (*string pattern* , *string string* , *array regs*)

mb_ereg utilise l'expression rationnelle pattern avec le support des caractères multi-octets sur la chaîne string . mb_ereg retourne 1 si des segments de chaîne qui vérifient le masque ont été trouvés. Le troisième paramètre est optionnel. S'il est fourni, mb_ereg retournera la taille du segment de chaîne identifié, regs contient les sous-chaînes. Si rien n'est trouvé, la fonction retourne FALSE .

Le jeu de caractères interne, ou le jeu de caractères spécifié par mb_regex_encoding sera utilisé.

Voir aussi mb_regex_encoding et mb_eregi .

8.67.32 mb_eregi_replace() : Expression rationnelle avec support des caractères multi-octets, sans tenir compte de la casse

string **mb_eregi_replace** (*string pattern* , *string replace* , *string string* , *string option*)

mb_eregi_replace analyse la chaîne string avec le masque d'expression rationnelle pattern , puis remplace le texte trouvé par remplacement . Enfin, mb_eregi_replace retourne la nouvelle chaîne ainsi bâtie, ou FALSE en cas d'erreur. Les caractères multi-octets peuvent être utilisés dans la chaîne pattern . La casse sera ignorée. Le paramètre option a la même signification que pour la fonction mb_ereg_replace .

Le jeu de caractères interne, ou le jeu de caractères est spécifié par mb_regex_encoding .

Voir aussi mb_regex_encoding et mb_ereg_replace .

8.67.29 mb_ereg_search_setpos() : Choisit le point de départ de la recherche par expression rationnelle

8.67.33 **mb_eregi()** : Expression rationnelle insensible à la casse avec le support des caractères multi-octets

int **mb_eregi** (string pattern , string string , array regs)

mb_eregi utilise l'expression rationnelle *pattern* avec le support des caractères multi-octets sur la chaîne *string* . **mb_eregi** ignore la casse dans ses recherches. **mb_eregi** retourne 1 si des segments de chaîne qui vérifient le masque ont été trouvés. Le troisième paramètre est optionnel. S'il est fourni, **mb_eregi** retournera la taille du segment de chaîne identifié, *regs* contient les sous-chaînes. Si rien n'est trouvé, la fonction retourne FALSE .

Le jeu de caractères interne, ou le jeu de caractères est spécifié par [mb_regex_encoding](#) .

Voir aussi [mb_regex_encoding](#) et [mb_ereg](#) .

8.67.34 **mb_get_info()** : Lit la configuration interne de l'extension mbstring

mixed **mb_get_info** (string type)

mb_get_info lit la configuration interne de l'extension mbstring internal setting.

Si *type* n'est pas spécifié, ou s'il vaut "all", un tableau contenant les éléments "internal_encoding", "http_output", "http_input" et "func_overload" sera retourné.

Si *type* est spécifié, et s'il vaut "http_output", "http_input", "internal_encoding" ou "func_overload", la paramètre spécifié sera retourné.

Voir aussi [mb_internal_encoding](#) et [mb_http_output](#) .

8.67.35 **mb_http_input()** : Détecte le type d'encodage d'un caractère HTTP

mixed **mb_http_input** (string type)

mb_http_input retourne le type d'encodage utilisé par une requête HTTP.

Le paramètre *type* spécifie le type d'entrée HTTP. Il peut prendre l'une des valeurs suivantes : "G" pour GET, "P" pour POST, "C" pour COOKIE, "S" pour chaîne de caractères, "L" pour la liste, "I" pour la liste complète (retournera un tableau). Si *type* est omis, il prend la valeur du dernier type utilisé.

Valeur retournée : nom de l'encodage utilisé. Si **mb_http_input** ne peut traiter ce type d'encodage, elle retourne FALSE .

Voir aussi [mb_internal_encoding](#) , [mb_http_output](#) et [mb_detect_order](#)

8.67.36 `mb_http_output()` : Lit/modifie l'encodage d'affichage

mixed `mb_http_output` (*string encoding*)

Si `encoding` est fourni, `mb_http_output` utilisera dorénavant l'encodage `encoding` pour les affichages HTTP : les caractères qui seront envoyés aux clients web seront convertis dans le jeu de caractères `encoding`. `mb_http_output` retourne TRUE en cas de succès, et FALSE en cas d'échec.

Si `encoding` est omis, `mb_http_output` retourne l'encodage d'affichage courant.

Voir aussi `mb_internal_encoding`, `mb_http_input` et `mb_detect_order`

8.67.37 `mb_internal_encoding()` : Lit/modifie l'encodage interne

mixed `mb_internal_encoding` (*string encoding*)

`mb_internal_encoding` modifie l'encodage interne courant en le remplaçant par `encoding`. Si ce paramètre est omis, l'encodage interne courant est retourné.

`encoding` sert lors des conversions des chaînes en provenance et en direction du web, ainsi que lors de la création de chaînes avec le module `mbstring`.

`encoding` : Nom d'encodage.

Valeur retournée : si `encoding` est fourni, `mb_internal_encoding` retourne TRUE en cas de succès, et FALSE sinon. Si `encoding` est omis, `mb_internal_encoding` retourne le nom de l'encodage courant.

Exemple avec `mb_internal_encoding`

```

<?php
/* Utilise l'encodage interne UTF-8 */
mb_internal_encoding("UTF-8");

/* Affiche l'encodage interne courant */
echo mb_internal_encoding();
?>

```

Voir aussi `mb_http_input`, `mb_http_output` et `mb_detect_order`

8.67.38 `mb_language()` : Lit/modifie le langage courant

mixed `mb_language` (*string language*)

`mb_language` remplace le langage courant par `language`. Si `language` est omis, `mb_language` retourne le langage courant.

Le paramètre `language` sert à encoder les messages électroniques. Les langages valides sont : "Japanese" (japonais), "ja" (japonais), "English" (anglais), "en" (anglais) et "uni" (UTF-8).

`mb_send_mail` utilise cette option pour encoder les emails.

Le langage et sa configuration valent ISO-2022-JP/Base64 pour le japonais, UTF-8/Base64 pour l'UTF-8 et ISO-8859-1/quoted printable pour l'anglais.

Si language est fourni et est valide, `mb_language` retourne TRUE . Sinon, elle retourne FALSE . Lorsque le paramètre language est omis, `mb_language` retourne le nom du langage courant, sous forme de chaîne. Si aucun langage n'a été configuré, `mb_language` retourne FALSE .

Voir aussi `mb_send_mail` .

8.67.39 `mb_list_encodings()` : Retourne un tableau contenant tous les encodages supportés

array `mb_list_encodings` (void)

`mb_list_encodings` retourne un tableau contenant tous les encodages supportés.

Exemple avec `mb_list_encodings`

```
<?php
print_r(mb_list_encodings());
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[0] => pass
[1] => auto
[2] => wchar
[3] => byte2be
[4] => byte2le
[5] => byte4be
[6] => byte4le
[7] => BASE64
[8] => UUENCODE
[9] => HTML-ENTITIES
[10] => Quoted-Printable
[11] => 7bit
[12] => 8bit
[13] => UCS-4
[14] => UCS-4BE
[15] => UCS-4LE
[16] => UCS-2
[17] => UCS-2BE
[18] => UCS-2LE
[19] => UTF-32
[20] => UTF-32BE
[21] => UTF-32LE
[22] => UTF-16
[23] => UTF-16BE
[24] => UTF-16LE
[25] => UTF-8
[26] => UTF-7
[27] => UTF7-IMAP
```

```
[28] => ASCII
[29] => EUC-JP
[30] => SJIS
[31] => eucJP-win
[32] => SJIS-win
[33] => JIS
[34] => ISO-2022-JP
[35] => Windows-1252
[36] => ISO-8859-1
[37] => ISO-8859-2
[38] => ISO-8859-3
[39] => ISO-8859-4
[40] => ISO-8859-5
[41] => ISO-8859-6
[42] => ISO-8859-7
[43] => ISO-8859-8
[44] => ISO-8859-9
[45] => ISO-8859-10
[46] => ISO-8859-13
[47] => ISO-8859-14
[48] => ISO-8859-15
[49] => EUC-CN
[50] => CP936
[51] => HZ
[52] => EUC-TW
[53] => BIG-5
[54] => EUC-KR
[55] => UHC
[56] => ISO-2022-KR
[57] => Windows-1251
[58] => CP866
[59] => KOI8-R
)
```

8.67.40 mb_output_handler() : Fonction de traitement des affichages

string **mb_output_handler** (string contents , int status)

mb_output_handler est la fonction à fournir à ob_start . mb_output_handler convertit les caractères envoyés au client dans l'encodage paramétré avec mb_http_output .

Dans les versions 4.1.0 de PHP ou supérieures, cette fonction ajoute un en-tête HTTP de jeu de caractères, en suivant ces conditions :

- Content-Type n'a pas été défini par la fonction header
- Le type MIME par défaut commence alors par text/
- La configuration http_output est différente de pass

contents : le contenu à traiter

status : l'état du contenu

mb_output_handler retourne la chaîne convertie.

Exemple avec `mb_output_handler`

```
<?php
  mb_http_output ("UTF-8");
  ob_start ("mb_output_handler");
?>
```

Note

Si vous souhaitez envoyer des données binaires telles des images issues d'un script PHP 4.3.0 ou suivante, Content-Type: header doit être envoyé en utilisant la fonction `header` avant d'envoyer les données binaires au client (e.g. `header("Content-Type: image/png")`). Si Content-Type: header est envoyé, la conversion de l'encodage de sortie ne sera pas effectuée.

Notez que si Content-Type: text/* est envoyé en utilisant la fonction `header`, les données envoyées seront considérées comme du texte, et la conversion sera effectuée en utilisant le jeu de caractères associé.

Si vous souhaitez envoyer des données binaires telles des images issues d'un script PHP, vous devez spécifier l'encodage spécial "pass", avec la fonction `mb_http_output`.

Voir aussi `ob_start`.

8.67.41 `mb_parse_str()` : Analyse les données HTTP GET/POST/COOKIE et assigne les variables globales

`bool mb_parse_str (string encoded_string , array result)`

`mb_parse_str` analyse les données d'entrées HTTP GET/POST/COOKIE et assigne les variables globales. Etant donné que PHP ne fournit pas de valeurs brutes de POST/COOKIE, cette fonction n'est utilisable que sur les données en méthode GET. `mb_parse_str` prend les données de l'URL appelante, détecte le jeu de caractères, convertit les données en jeu de caractères interne, et affecte les valeurs au tableau de variables globales.

`encoded_string` : Les données encodées de l'URL.

`result` : un tableau contenant les valeurs décodées, et les noms des jeux de caractères.

`mb_parse_str` retourne TRUE en cas de succès, et FALSE sinon.

Voir aussi `mb_detect_order` et `mb_internal_encoding`.

8.67.42 `mb_preferred_mime_name()` : Détecte l'encodage MIME

`string mb_preferred_mime_name (string encoding)`

`mb_preferred_mime_name` retourne le type d'encodage MIME utilisé dans le mail encoding. Le nom de l'encodage est retourné sous forme de chaîne.

Exemple avec `mb_preferred_mime_string`

```
<?php
  $outputenc = "sjis-win";
  mb_http_output ($outputenc);
  ob_start ("mb_output_handler");
  header ("Content-Type: text/html; charset=" . mb_preferred_mime_name ($outputenc));
?>
```

8.67.43 `mb_regex_encoding()` : Retourne le jeu de caractères courant pour les expressions rationnelles

mixed `mb_regex_encoding` (*string encoding*)

`mb_regex_encoding` retourne le jeu de caractères courant pour les expressions rationnelles, sous forme de chaîne.

Si le troisième argument optionnel `encoding` est fourni, il prendra la valeur du jeu de caractères par défaut pour les expressions rationnelles. Par défaut, cette option est identique au jeu de caractères interne.

Voir aussi `mb_internal_encoding` et `mb_ereg` .

8.67.44 `mb_regex_set_options()` : Lit et modifie les options des fonctions d'expression rationnelle à support de caractères multi-octets

string `mb_regex_set_options` (*string options*)

`mb_regex_set_options` configure les options par défaut avec les nouvelles valeurs contenues dans `options` , pour les fonctions d'expression rationnelle à support de caractères multi-octets.

`mb_regex_set_options` retourne la configuration précédente. Si `options` est omise, la fonction retourne la chaîne qui décrit les options courantes.

Voir aussi `mb_split` , `mb_ereg` et `mb_eregi` .

8.67.45 `mb_send_mail()` : Envoie un mail encodé

bool `mb_send_mail` (string `to` , string `subject` , string `message` , *string additional_headers* , *string additional_parameter*)

`mb_send_mail` envoie un courrier électronique. Les en-têtes et le corps du message sont convertis et encodés en accord avec `mb_language` . `mb_send_mail` est une version adaptée de `mail` . Voir la fonction `mail` pour plus de détails.

`to` est l'adresse de destination du mail. Les adresses multiples peuvent être spécifiées en les séparant par des virgules. Ce paramètre n'est pas encodé automatiquement.

`subject` est le sujet du mail.

`message` est le message du mail.

La chaîne `additional_headers` est insérée à la fin de l'en-tête mail. Elle sert à ajouter d'autres en-têtes email. N'oubliez pas de les séparer par des nouvelles lignes (`\n`).

Note

Les en-têtes `Content-Type` et `Content-Transfer-Encoding` peuvent être redéfinis depuis PHP 5.0.0. En PHP 4, les valeurs définies par la fonction `mb_language` sont toujours utilisées.

`additional_parameter` est une ligne de paramètres MTA. Il est pratique lorsque vous voulez définir un Return-Path correct lorsque vous utilisez `sendmail` .

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Voir aussi [mail](#) , [mb_encode_mimeheader](#) et [mb_language](#) .

8.67.46 `mb_split()` : Scinde une chaîne en tableau avec une expression rationnelle multi-octets

array `mb_split` (string `pattern` , string `string` , int `limit`)

`mb_split` scinde la chaîne multi-octets `string` en utilisant l'expression rationnelle `pattern` puis en retournant le résultat sous forme de tableau.

Si le paramètre optionnel `limit` est spécifié, la chaîne sera scindée en `limit` éléments au plus.

Le jeu de caractères interne, ou le jeu de caractères est spécifié par [mb_regex_encoding](#) .

Voir aussi [mb_regex_encoding](#) et [mb_ereg](#) .

8.67.47 `mb_strcut()` : Coupe une partie de chaîne

string `mb_strcut` (string `str` , int `start` , int `length` , string `encoding`)

`mb_strcut` retourne la portion de la chaîne `str` qui commence au caractère `start` et a la longueur de `length` caractères.

`mb_strcut` effectue une opération équivalente à `mb_substr` avec des méthodes différentes. Si la position `start` est un caractère multi-octets ou plus, il débutera à partir du premier octet ou du premier caractère multi-octets.

Il soustrait la chaîne `str` qui est plus courte que `length` ET caractères qui ne font pas partis d'une chaîne multi-octets et qui ne commence pas au milieu de la séquence.

`encoding` est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Voir aussi [mb_substr](#) et [mb_internal_encoding](#) .

8.67.48 `mb_strimwidth()` : Tronque une chaîne

string `mb_strimwidth` (string `str` , int `start` , int `width` , string `trimmarker` , string `encoding`)

`mb_strimwidth` tronque la chaîne `str` à la longueur `width` . Elle retourne la chaîne tronquée.

Si `trimmarker` est fourni, `trimmarker` est ajoutée à la fin de la chaîne retournée.

`start` est l'offset de départ, en nombre de caractères depuis le début de la chaîne (cela commence à 0).

trimmarker est la chaîne ajoutée à la fin de la chaîne tronquée.

encoding est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Exemple avec `mb_strimwidth`

```
<?php
$str = mb_strimwidth($str, 0, 40, "...");
?>
```

Voir aussi [mb_strwidth](#) et [mb_internal_encoding](#) .

8.67.49 `mb_strlen()` : Retourne la taille d'une chaîne

int `mb_strlen` (string *str* , *string encoding*)

`mb_strlen` retourne le nombre de caractères dans la chaîne *str* , avec l'encodage *encoding* . Un caractère multi-octets est alors compté pour 1.

encoding est l'encodage de *str* . Si *encoding* est omis, l'encodage interne sera utilisé.

Voir aussi [mb_internal_encoding](#) , [strlen](#) .

8.67.50 `mb_strpos()` : Repère la première occurrence d'un caractère dans une chaîne

int `mb_strpos` (string *haystack* , string *needle* , *int offset* , *string encoding*)

`mb_strpos` retourne la position numérique de la première occurrence du caractère *needle* dans la chaîne *haystack* . Si *needle* est introuvable, `mb_strpos` retourne FALSE .

`mb_strpos` effectue une recherche de type `strpos` , en tenant compte des caractères multi-octets. La position de *needle* est comptée à partir du début de la chaîne *haystack* : les positions commencent à 0.

Si *encoding* est omis, l'encodage interne par défaut est utilisé. `mb_strpos` accepte des chaînes comme argument *needle* , alors que `strpos` n'accepte que des caractères.

offset est l'offset de début de recherche. S'il est omis, zéro sera utilisé (début de la chaîne).

encoding est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Voir aussi [mb_strrpos](#) , [mb_internal_encoding](#) et [strpos](#) .

8.67.51 `mb_strrpos()` : Repère la dernière occurrence d'un caractère dans une chaîne

int `mb_strrpos` (string *haystack* , string *needle* , *string encoding*)

`mb_strrpos` retourne la position numérique de la dernière occurrence du caractère *needle* dans la chaîne *haystack* . Si *needle* est introuvable, `mb_strrpos` retourne FALSE .

`mb_strrpos` effectue une recherche de type `strrpos`, en tenant compte des caractères multi-octets. La position de `needle` est comptée à partir du début de la chaîne haystack : les positions commencent à 0.

Si `encoding` est omis, l'encodage interne par défaut est utilisé. `mb_strrpos` accepte des chaînes comme argument `needle`, alors que `strrpos` n'accepte que des caractères.

`encoding` est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Voir aussi `mb_strpos`, `mb_internal_encoding` et `strrpos`.

8.67.52 `mb_strtolower()` : Met tous les caractères en minuscules

string `mb_strtolower` (string `str`, string `encoding`)

`mb_strtolower` retourne la chaîne `str` après avoir converti tous les caractères alphabétiques en minuscules.

`encoding` spécifie le jeu de caractères de la chaîne `str`; s'il est omis, le jeu de caractères interne sera utilisé.

Pour plus d'informations sur les propriétés de l'Unicode, voyez

<http://www.unicode.org/unicode/reports/tr21/>.

Contrairement à `strtolower`, le concept de caractère 'alphabétique' est déterminé par les propriétés Unicode. De ce fait, le comportement de cette fonction n'est pas modifié par les configurations locales, et elle peut convertir tout les caractères qui sont considérés comme alphabétiques comme le c cédille (Ç).

Exemple avec `mb_strtolower`

```
<?php
$str = "Marie A Un Petit Agneau Et Elle L'Aime BEAUCOUP.";
$str = mb_strtolower($str);
echo $str; // marie a un petit agneau et elle l'aime beaucoup
?>
```

Voir aussi `strtolower`, `mb_strtoupper` et `mb_convert_case`.

8.67.53 `mb_strtoupper()` : Met tous les caractères en majuscules

string `mb_strtoupper` (string `str`, string `encoding`)

`mb_strtoupper` retourne la chaîne `str` après avoir converti tous les caractères alphabétiques en majuscules.

`encoding` spécifie le jeu de caractères de la chaîne `str`; s'il est omis, le jeu de caractères interne sera utilisé.

Pour plus d'informations sur les propriétés de l'Unicode, voyez

<http://www.unicode.org/unicode/reports/tr21/>.

Contrairement à [strtoupper](#) , le concept de caractère 'alphabétique' est déterminé par les propriétés Unicode. De ce fait, le comportement de cette fonction n'est pas modifié par les configurations locales, et elle peut convertir tout les caractères qui sont considérés comme alphabétiques comme le c cédille (ç).

Exemple avec [mb_strtoupper](#)

```
<?php
$str = "Marie A Un Petit Agneau Et Elle L'Aime BEAUCOUP.";
$str = mb_strtoupper($str);
echo $str; // MARIE A UN PETIT AGNEAU ET ELLE L'AIME BEAUCOUP.
?>
```

Voir aussi [strtoupper](#) , [mb_strtolower](#) et [mb_convert_case](#) .

8.67.54 [mb_strwidth\(\)](#) : Retourne la largeur d'une chaîne

int [mb_strwidth](#) (string *str* , string *encoding*)

[mb_strwidth](#) retourne la largeur de la chaîne *str* .

Les chaînes à encodage multi-octets sont généralement deux fois plus grandes que les chaînes à simple-octet.

Caractères	Taille
U+0000 - U+0019	0
U+0020 - U+1FFF	1
U+2000 - U+FF60	2
U+FF61 - U+FF9F	1
U+FFA0 -	2

encoding est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Voir aussi [mb_strimwidth](#) et [mb_internal_encoding](#) .

8.67.55 [mb_substitute_character\(\)](#) : Lit/modifie les caractères de substitution

mixed [mb_substitute_character](#) (mixed *substrchar*)

[mb_substitute_character](#) spécifie le caractère de substitution des caractères invalides, ou des encodages invalides. Les caractères invalides peuvent être remplacés par NULL (pas d'affichage, ils sont supprimés), une chaîne ou un code hexadécimal.

Ce paramétrage affecte [mb_convert_encoding](#) , [mb_convert_variables](#) , [mb_output_handler](#) , et [mb_send_mail](#) .

substchar spécifie une valeur Unicode sous la forme d'un entier, ou bien une chaîne sous ces formes :

- "none" : pas d'affichage

- "long" : affiche la valeur hexadécimale (par exemple : U+3000,JIS+7E7E)

Si substchar est fourni, `mb_substitute_character` retourne TRUE en cas de succès, et FALSE en cas d'erreur. Si substchar est omis, `mb_substitute_character` retourne une valeur Unicode, ou bien "none"/"long".

Exemple avec `mb_substitute_character`

```
<?php
/* Configure le caractère de substitution avec U+3013 (GETA MARK) */
mb_substitute_character(0x3013);

/* Configure le caractère de substitution avec un format hexadécimal */
mb_substitute_character("long");

/* Affiche la configuration courante */
echo mb_substitute_character();
?>
```

8.67.56 `mb_substr_count()` : Compte le nombre d'occurrences d'une sous-chaîne

int `mb_substr_count` (string haystack , string needle , *string encoding*)

`mb_substr_count` retourne le nombre d'occurrences de la chaîne needle dans la chaîne haystack string.

encoding spécifie le jeu de caractères utilisé pour needle et haystack . Si omis, le jeu de caractères interne est utilisé.

Exemple avec `mb_substr_count`

```
<?php
echo mb_substr_count("Ceci est un test", "es"); // affiche 2
?>
```

Voir aussi `substr_count` , `mb_strpos` et `mb_substr` .

8.67.57 `mb_substr()` : Lit une sous-chaîne

string `mb_substr` (string str , int start , *int length* , *string encoding*)

`mb_substr` retourne la portion de la chaîne str qui commence au caractère start et a la longueur de length caractères.

`mb_substr` effectue une recherche de type `strpos` , en tenant compte des caractères multi-octets. La position de needle est comptée à partir du début de la chaîne haystack : les positions commencent à 0.

Si encoding est omis, l'encodage interne est utilisé.

encoding est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Voir aussi `mb_strcut` et `mb_internal_encoding` .

8.68 MCAL

8.68.1 Introduction

MCAL est le sigle de Modular Calendar Access Library (bibliothèque calendaire modulaire).

Libmcal est une bibliothèque C de calendriers. Elle est écrite pour être très modulaire, et dispose de nombreux modules. MCAL est l'équivalent de IMAP pour les calendriers.

Avec mcal, un calendrier peut être ouvert comme une boîte aux lettres. Les calendriers peuvent être des fichiers locaux, ou bien être sur des serveurs ICAP distants, ou encore tout autre format supporté par la bibliothèque.

Les événements peuvent être lus, sélectionnés et enregistrés. Il y a aussi la possibilité d'ajouter des alarmes, et de placer des événements récurrents.

Avec libmcal, les serveurs centralisés peuvent être accédés et utilisés, et remplacent avantageusement tout développement spécifique de base de données.

La plupart des fonctions utilisent une structure d'événement interne, qui est unique pour chaque connexion. Cela évite d'avoir à passer des objets de grande taille entre les fonctions. Il existe des fonctions bien pratiques pour créer, initialiser et lire des objets événements.

Note
Cette extension a été déplacée dans le module <u>PECL</u> et ne sera plus intégrée dans PHP à partir de PHP 5.0.0.
Note
PHP disposait d'une extension ICAP, mais la bibliothèque originale et l'extension PHP ne sont plus supportées. La solution de remplacement est MCAL.
Note
Cette extension n'est pas disponible sur les plates-formes Windows.

8.68.2 Pré-requis

Pour faire fonctionner cette bibliothèque, vous devez compiler PHP avec l'option `--with-mcal`. Il vous faudra alors avoir installé la bibliothèque mcal. Téléchargez la dernière version à <http://mcal.chek.com/> et compilez-la, puis installez-la.

8.68.3 Installation

Après avoir installé la bibliothèque mcal, vous devez compiler PHP avec l'option `--with-mcal[=DIR]`.

8.68.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.68.5 Types de ressources

Cette extension ne définit aucune ressource.

8.68.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

MCAL_SUNDAY ([entier](#))
MCAL_MONDAY ([entier](#))
MCAL_TUESDAY ([entier](#))
MCAL_WEDNESDAY ([entier](#))
MCAL_THURSDAY ([entier](#))
MCAL_FRIDAY ([entier](#))
MCAL_SATURDAY ([entier](#))
MCAL_JANUARY ([entier](#))
MCAL_FEBRUARY ([entier](#))
MCAL_MARCH ([entier](#))
MCAL_APRIL ([entier](#))
MCAL_MAY ([entier](#))
MCAL_JUNE ([entier](#))
MCAL_JULY ([entier](#))
MCAL_AUGUST ([entier](#))
MCAL_SEPTEMBER ([entier](#))
MCAL_OCTOBER ([entier](#))
MCAL_NOVEMBER ([entier](#))
MCAL_DECEMBER ([entier](#))
MCAL_RECUR_NONE ([entier](#))
MCAL_RECUR_DAILY ([entier](#))
MCAL_RECUR_WEEKLY ([entier](#))
MCAL_RECUR_MONTHLY_MDAY ([entier](#))
MCAL_RECUR_MONTHLY_WDAY ([entier](#))
MCAL_RECUR_YEARLY ([entier](#))
MCAL_M_SUNDAY ([entier](#))
MCAL_M_MONDAY ([entier](#))
MCAL_M_TUESDAY ([entier](#))
MCAL_M_WEDNESDAY ([entier](#))
MCAL_M_THURSDAY ([entier](#))
MCAL_M_FRIDAY ([entier](#))
MCAL_M_SATURDAY ([entier](#))
MCAL_M_WEEKDAYS ([entier](#))
MCAL_M_WEEKEND ([entier](#))
MCAL_M_ALLDAYS ([entier](#))

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [mcal_append_event](#)
- [mcal_close](#)

- [mcal_create_calendar](#)
- [mcal_date_compare](#)
- [mcal_date_valid](#)
- [mcal_day_of_week](#)
- [mcal_day_of_year](#)
- [mcal_days_in_month](#)
- [mcal_delete_calendar](#)
- [mcal_delete_event](#)
- [mcal_event_add_attribute](#)
- [mcal_event_init](#)
- [mcal_event_set_alarm](#)
- [mcal_event_set_category](#)
- [mcal_event_set_class](#)
- [mcal_event_set_description](#)
- [mcal_event_set_end](#)
- [mcal_event_set_recur_daily](#)
- [mcal_event_set_recur_monthly_mday](#)
- [mcal_event_set_recur_monthly_wday](#)
- [mcal_event_set_recur_none](#)
- [mcal_event_set_recur_weekly](#)
- [mcal_event_set_recur_yearly](#)
- [mcal_event_set_start](#)
- [mcal_event_set_title](#)
- [mcal_expunge](#)
- [mcal_fetch_current_stream_event](#)
- [mcal_fetch_event](#)
- [mcal_is_leap_year](#)
- [mcal_list_alarms](#)
- [mcal_list_events](#)
- [mcal_next_recurrence](#)
- [mcal_open](#)
- [mcal_popen](#)
- [mcal_rename_calendar](#)
- [mcal_reopen](#)
- [mcal_snooze](#)
- [mcal_store_event](#)
- [mcal_time_valid](#)
- [mcal_week_of_year](#)

8.68.8 [mcal_close\(\)](#) : Ferme une connexion MCAL

`bool mcal_close (int mcal_stream , int flags)`

[mcal_close](#) ferme la connexion [mcal_stream](#) .

8.68.9 [mcal_create_calendar\(\)](#) : Crée un nouveau calendrier

`bool mcal_create_calendar (int stream , string calendar)`

[mcal_create_calendar](#) crée un nouveau calendrier nommé [calendar](#) , sur le serveur MCAL [stream](#) .

8.68.10 `mcal_date_compare()` : Compare deux dates

int `mcal_date_compare` (int `a_year` , int `a_month` , int `a_day` , int `b_year` , int `b_month` , int `b_day`)

`mcal_date_compare` compare les deux dates données, et retourne <0, 0, > 0 si a<b, a==b, a>b respectivement.

8.68.11 `mcal_date_valid()` : Valide une date

bool `mcal_date_valid` (int `year` , int `month` , int `day`)

`mcal_date_valid` retourne TRUE si la date (constituée par l'année `year` , le mois `month` et le jour `day`) est valide, et FALSE sinon.

8.68.12 `mcal_day_of_week()` : Le jour de la semaine

int `mcal_day_of_week` (int `year` , int `month` , int `day`)

`mcal_day_of_week` retourne le jour de la semaine, pour la date constituée par l'année `year` , le mois `month` et le jour `day` . Les valeurs retournées seront dans l'intervalle 0-6, avec 0 correspondant à Dimanche et 6 à Samedi.

8.68.13 `mcal_day_of_year()` : Le jour de l'année

int `mcal_day_of_year` (int `year` , int `month` , int `day`)

`mcal_day_of_year` retourne le numéro de jour dans l'année pour la date constituée par l'année `year` , le mois `month` et le jour `day` .

8.68.14 `mcal_days_in_month()` : Retourne le nombre de jours dans un mois

int `mcal_days_in_month` (int `month` , int `leap_year`)

`mcal_days_in_month` retourne le nombre de jours du mois `month` , et prend en compte le fait que l'année est bissextile avec le paramètre `leap_year` .

8.68.15 `mcal_delete_calendar()` : Efface un calendrier

bool `mcal_delete_calendar` (int `stream` , string `calendar`)

`mcal_delete_calendar` efface le calendrier `calendar` sur le serveur `stream` .

8.68.16 mcal_delete_event() : Efface un événement dans un calendrier MCAL

```
bool mcal_delete_event ( int mcal_stream , int event_id )
```

mcal_delete_event efface l'événement dont l'identifiant est uid .

mcal_delete_event retourne TRUE .

8.68.17 mcal_event_add_attribute() : Ajoute un attribut et une valeur à la structure globale

```
bool mcal_event_add_attribute ( int stream , string attribute , string value )
```

mcal_event_add_attribute ajoute l'attribut attribute à la structure globale, avec la valeur value .

8.68.18 mcal_event_init() : Initialise la structure globale d'un flux MCAL

```
void mcal_event_init ( int stream )
```

mcal_event_init initialise la structure globale d'un flux mcal. Cela remet tous les éléments de la structure à 0, ou à leur valeur par défaut.

8.68.19 mcal_event_set_alarm() : Fixe l'alarme de la structure globale

```
void mcal_event_set_alarm ( int stream , int alarm )
```

mcal_event_set_alarm fixe l'alarme de la structure globale à un nombre de minutes avant déclenchement.

8.68.20 mcal_event_set_category() : Fixe la catégorie de la structure globale

```
void mcal_event_set_category ( int stream , string category )
```

mcal_event_set_category fixe la catégorie de la structure globale à la valeur de category .

8.68.21 mcal_event_set_class() : Fixe la classe de la structure globale

```
void mcal_event_set_class ( int stream , int class )
```

mcal_event_set_class fixe la classe de la structure globale. La classe vaut 1 si elle est publique, et 0 si elle est privée.

8.68.22 mcal_event_set_description() : Fixe la description de la structure globale

```
void mcal_event_set_description ( int stream , string description )
```

mcal_event_set_description fixe la catégorie de la structure globale à la valeur de description .

8.68.23 mcal_event_set_end() : Fixe la date de fin de la structure globale

```
void mcal_event_set_end ( int stream , int year , int month , int day , int hour , int min , int sec )
```

mcal_event_set_end fixe la date de fin de la structure globale.

8.68.24 mcal_event_set_recur_daily() : Fixe la récurrence quotidienne

```
void mcal_event_set_recur_daily ( int stream , int year , int month , int day , int interval )
```

mcal_event_set_recur_daily fixe la récurrence quotidienne de la structure globale, jusqu'à la date passée en paramètre (la date de début est celle de la structure).

8.68.25 mcal_event_set_recur_monthly_mday() : Fixe la récurrence

```
void mcal_event_set_recur_monthly_mday ( int stream , int year , int month , int day , int interval )
```

mcal_event_set_recur_monthly_mday fixe la récurrence de la structure globale, jusqu'à la date passée en paramètre (la date de début est celle de la structure).

8.68.26 mcal_event_set_recur_monthly_wday() : Fixe la récurrence mensuelle

```
void mcal_event_set_recur_monthly_wday ( int stream , int year , int month , int day , int interval )
```

mcal_event_set_recur_monthly_wday fixe la récurrence mensuelle de la structure globale, jusqu'à la date passée en paramètre. (La date de début est celle de la structure).

8.68.27 mcal_event_set_recur_none() : Supprime la récurrence de la structure globale

```
void mcal_event_set_recur_none ( int stream )
```

mcal_event_set_recur_none supprime la récurrence de la structure globale (event->recur_type est mis à MCAL_RECUR_NONE).

8.68.28 mcal_event_set_recur_weekly() : Fixe la récurrence hebdomadaire

void **mcal_event_set_recur_weekly** (int stream , int year , int month , int day , int interval , int weekdays)

mcal_event_set_recur_weekly fixe la récurrence hebdomadaire de la structure globale, jusqu'à la date passée en paramètre. (la date de début est celle de la structure).

8.68.29 mcal_event_set_recur_yearly() : Fixe la récurrence annuelle

void **mcal_event_set_recur_yearly** (int stream , int year , int month , int day , int interval)

mcal_event_set_recur_yearly fixe la récurrence annuelle de la structure globale, jusqu'à la date passée en paramètre (la date de début est celle de la structure).

8.68.30 mcal_event_set_start() : Fixe la date de début de la structure globale

void **mcal_event_set_start** (int stream , int year , int month , int day , int hour , int min , int sec)

mcal_event_set_start fixe la date de début de la structure globale.

8.68.31 mcal_event_set_title() : Fixe le titre de la structure globale

void **mcal_event_set_title** (int stream , string title)

mcal_event_set_title fixe le titre de la structure globale à la valeur de title .

8.68.32 mcal_expunge() : Supprime tous les événements marqués pour l'effacement

bool **mcal_expunge** (int stream)

mcal_expunge supprime tous les événements marqués pour l'effacement sur la connexion stream .

8.68.33 mcal_fetch_current_stream_event() : Retourne un objet contenant la structure de date

object **mcal_fetch_current_stream_event** (int stream)

mcal_fetch_current_stream_event retourne la structure de la date du flux courant sous la forme d'un objet, qui contient :

- int id - ID de l'événement.
- int public - TRUE si l'événement est public, FALSE s'il est privé.

- string category - Catégorie de l'événement.
- string title - Titre de l'événement.
- string description - Description de l'événement.
- int alarm - Nombre de minutes avant d'envoyer une alerte pour cet événement.
- object start - Objet contenant une date et une heure.
- object end - Objet contenant une date et une heure.
- int recur_type - type de récurrence
- int recur_interval - intervalle de récurrence
- datetime recur_endate - date de fin de récurrence
- int recur_data - données de récurrence

Tous les objets de date et heure sont construits comme suit :

- int year - année
- int month - mois
- int mday - jour du mois
- int hour - heure
- int min - minutes
- int sec - secondes
- int alarm - nombre de minutes avant de déclencher l'alarme

8.68.34 `mcalfetch_event()` : Recherche un événement dans le calendrier

objet `mcalfetch_event` (int `mcalfetch_stream` , int `event_id` , int `options`)

`mcalfetch_event` recherche un événement dans le calendrier spécifié par id .

`mcalfetch_event` retourne un objet événement dont les attributs sont :

- int id - ID de l'événement.
- int public - TRUE si l'événement est public, FALSE s'il est privé.
- string category - Catégorie de l'événement.
- string title - Titre de l'événement.
- string description - Description de l'événement.
- int alarm - Nombre de minutes avant d'envoyer une alerte pour cet événement.
- object start - Objet contenant une date et une heure.
- object end - Objet contenant une date et une heure.
- int recur_type - type de récurrence
- int recur_interval - intervalle de récurrence
- datetime recur_endate - date de fin de récurrence
- int recur_data - données de récurrence

Tous les objets de date et heure sont construits comme suit :

- int year - année
- int month - mois
- int mday - jour du mois
- int hour - heure
- int min - minutes
- int sec - secondes
- int alarm - nombre de minutes avant de déclencher l'alarme

8.68.33 `mcalfetch_current_stream_event()` : Retourne un objet contenant la structure de date 1100

Les valeurs possibles de `recur_type` sont :

- 0 - Indique que l'événement ne se répète jamais
- 1 - Indique que l'événement se répète tous les jours
- 2 - Indique que l'événement se répète toutes les semaines
- 3 - Indique que l'événement se répète tous les mois, à la même date (le 10 du mois)
- 4 - Indique que l'événement se répète tous les mois, un certain jour (i.e., le troisième samedi du mois)
- 5 - Indique que l'événement se répète tous les ans

8.68.35 `mcal_is_leap_year()` : Vérifie que l'année est bissextile

`bool mcaldis_leap_year (int year)`

`mcal_is_leap_year` retourne 1 si l'année `year` est bissextile, et 0 sinon.

8.68.36 `mcal_list_alarms()` : Retourne une liste d'événements qui ont une alarme prévue à une date

`array mcaldlist_alarms (int mcaldstream , int begin_year , int begin_month , int begin_day , int end_year , int end_month , int end_day)`

`mcal_list_events` retourne un tableau d'identifiants d'événements, qui ont une alarme de prévue à la date `alarm_date` . Si seul le flux MCAL est donné, la date de début et de fin de la structure globale sera utilisée.

`mcal_list_events` prend une date, et retourne un tableau d'identifiants.

8.68.37 `mcal_list_events()` : Retourne une liste d'événements pour une date ou entre deux dates

`array mcaldlist_events (int mcaldstream , int begin_year , int begin_month , int begin_day , int end_year , int end_month , int end_day)`

`mcal_list_events` retourne un tableau d'identifiants d'événements, compris entre deux dates ou pour une date précise.

`mcal_list_events` prend une date de début et une date de fin. Un tableau d'identifiants est retourné.

8.68.38 `mcal_next_recurrence()` : Retourne la prochaine occurrence d'un événement

`object mcaldnext_recurrence (int stream , int weekstart , array next)`

`mcal_next_recurrence` retourne un objet contenant la prochaine date de l'événement, ou la date de l'événement suivant la date. `mcal_next_recurrence` retourne un objet `date` vide si l'événement n'a pas de prochaine occurrence, ou si quelque chose est invalide. Utilisez `weekstart` pour déterminer le premier jour.

8.68.39 `mcald_open()` : Ouvre une connexion MCAL

`int mcald_open (string calendar , string username , string password , int options)`

`mcald_open` retourne un flux MCAL en cas de succès, et FALSE en cas d'erreur.

`mcald_open` ouvre une connexion MCAL au serveur calendar . Si options est spécifié, passe aussi options à la boîte aux lettres (???) . La structure interne du flux MCAL est initialisée à la connexion.

8.68.40 `mcald_popen()` : Ouvre une connexion persistante MCAL

`int mcald_popen (string calendar , string username , string password , int options)`

`mcald_popen` retourne un flux MCAL en cas de succès, et FALSE sinon.

`mcald_popen` ouvre une connexion MCAL au serveur de calendrier calendar . Si les options options sont spécifiées, elles sont aussi passées à cette boîte aux lettres (???) . La structure interne du flux est aussi initialisée.

8.68.41 `mcald_rename_calendar()` : Renomme un calendrier

`bool mcald_rename_calendar (int stream , string old_name , string new_name)`

`mcald_rename_calendar` renomme le calendrier old_name en new_name .

8.68.42 `mcald_reopen()` : Rouvre une connexion MCAL

`bool mcald_reopen (int mcald_stream , string calendar , int options)`

`mcald_reopen` rouvre une connexion MCAL.

`mcald_reopen` rouvre une connexion MCAL avec le serveur calendar . Si les options options sont spécifiées, elles sont aussi passées à cette boîte aux lettres.

8.68.43 `mcald_snooze()` : Eteind l'alarme d'un événement

`bool mcald_snooze (int stream_id , int event_id)`

`mcald_snooze` éteint l'alarme de l'événement identifié par stream_id et event_id .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.68.44 `mcald_store_event()` : Modifie un événement dans un calendrier MCAL

`int mcald_store_event (int mcald_stream)`

`mcald_store_event` enregistre l'événement global dans le calendrier MCAL mcald_stream .

mcal_store_event retourne l'identifiant de l'événement modifié en cas de succès, et FALSE en cas d'erreur.

8.68.45 mcal_time_valid() : Valide une heure

bool **mcal_time_valid** (int hour , int minutes , int seconds)

mcal_time_valid retourne TRUE si l'heure (constituée par l'heure hour , les minutes minutes et les secondes seconds) est une heure valide, et FALSE sinon.

8.68.46 mcal_week_of_year() : Retourne le numéro de semaine dans l'année pour une date donnée

int **mcal_week_of_year** (int day , int month , int year)

mcal_week_of_year retourne le numéro de la semaine pour la date donnée par l'année year , le mois month et le jour day .

8.69 chiffrement mcrypt

8.69.1 Introduction

Ces fonctions permettent d'accéder à la bibliothèque mcrypt, qui dispose d'une grande variété d'algorithmes de chiffrement, tels DES, TripleDES, Blowfish (par défaut), 3-WAY, SAFER-SK64, SAFER-SK128, TWOFISH, TEA, RC2 et GOST en modes CBC, OFB, CFB et ECB. De plus, elles acceptent aussi RC6 et IDEA qui sont considérés comme "non-libres".

8.69.2 Pré-requis

Ces fonctions utilisent `mcrypt`. Pour utiliser cette bibliothèque, téléchargez le fichier `libmcrypt-x.x.tar.gz` sur <http://mcrypt.sourceforge.net/> et suivez les instructions d'installation fournies. Les utilisateurs Windows trouveront toutes les bibliothèques nécessaires sur <http://ftp.emini.dk/pub/php/win32/mcrypt/>.

Depuis PHP 5.0.0, vous avez besoin de la version 2.5.6 ou suivant de la bibliothèque libmcrypt.

Si vous compilez PHP avec la bibliothèque libmcrypt 2.4.x, les algorithmes suivants sont supportés : CAST, LOKI97, RIJNDAEL, SAFERPLUS, SERPENT ainsi que les chiffrements suivants : ENIGMA (chiffrement), PANAMA, RC4 et WAKE. Avec libmcrypt 2.4.x un autre mode de chiffrement est disponible : nOFB.

8.69.3 Installation

Vous devez compiler PHP avec l'option `--with-mcrypt=[DIR]` pour activer cette extension. DIR est le dossier d'installation de mcrypt. Assurez-vous de compiler libmcrypt avec l'option `--disable-posix-threads`.

8.69.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable	Historique
<code>mcrypt.algorithms_dir</code>	NULL	PHP_INI_ALL	Disponible depuis PHP 4.0.2.
<code>mcrypt.modes_dir</code>	NULL	PHP_INI_ALL	Disponible depuis PHP 4.0.2.

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#).

8.69.5 Types de ressources

Cette extension ne définit aucune ressource.

8.69.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Mcrypt peut opérer en 4 modes de chiffrement (CBC, OFB, CFB, et ECB). Si vous utilisez libmcrypt-2.4.x ou plus récent, les fonctions peuvent aussi opérer en mode nOFB et en mode STREAM. Nous allons présenter la technique d'utilisation de ces modes. Pour plus de références et de détails, reportez-vous au livre suivant : Applied Cryptography by Schneier (ISBN 0-471-11709-9).

- MCRYPT_MODE_ECB (electronic codebook) est prévu pour des données aléatoires, telles que des clés. Comme les données sont peu nombreuses et aléatoires, les inconvénients de l'ECB ont ici un effet négatif favorable.
- MCRYPT_MODE_CBC (cipher block chaining) est spécialement pratique avec les fichiers dont la sécurité ECB n'est pas suffisante.
- MCRYPT_MODE_CFB (cipher feedback) est la meilleure méthode pour chiffrer des flots d'octets, quand les octets doivent être cryptés un par un.
- MCRYPT_MODE_OFB (output feedback, in 8bit) est comparable à CFB, mais peut être utilisé lorsque des erreurs ne doivent pas être propagées.
- MCRYPT_MODE_NOFB (output feedback, in nbit) est comparable à OFB, mais plus sûr, car il opère avec la taille de blocs de l'algorithme.
- MCRYPT_MODE_STREAM est un mode supplémentaire, pour permettre l'utilisation d'algorithmes tels WAKE ou RC4.

Voici quelques autres modes et méthodes de compression :

MCRYPT_ENCRYPT (entier)
MCRYPT_DECRYPT (entier)
MCRYPT_DEV_RANDOM (entier)
MCRYPT_DEV_URANDOM (entier)
MCRYPT_RAND (entier)

8.69.7 Mcrypt ciphers

Voici une liste non exhaustive des modes de chiffrement de l'extension mcrypt. Pour disposer d'une liste complète des chiffrements supportés, voyez les définitions dans le fichier mcrypt.h . La règle générale avec l'API mcrypt-2.2.x API est que vous pouvez accéder au mode de chiffrement depuis PHP avec la constante MCRYPT_ciphernam. Avec la bibliothèque libmcrypt-2.4.x et libmcrypt-2.5.x, ces constantes fonctionnent toujours, mais il est possible de spécifier le nom du chiffrement dans une chaîne, lors de l'appel à mcrypt_module_open .

- MCRYPT_3DES
- MCRYPT_ARCFOUR_IV (libmcrypt > 2.4.x seulement)
- MCRYPT_ARCFOUR (libmcrypt > 2.4.x seulement)
- MCRYPT_BLOWFISH
- MCRYPT_CAST_128
- MCRYPT_CAST_256
- MCRYPT_CRYPT
- MCRYPT_DES
- MCRYPT_DES_COMPAT (libmcrypt 2.2.x seulement)
- MCRYPT_ENIGMA (libmcrypt > 2.4.x seulement, alias de MCRYPT_CRYPT)
- MCRYPT_GOST
- MCRYPT_IDEA (non-free)
- MCRYPT_LOKI97 (libmcrypt > 2.4.x seulement)
- MCRYPT_MARS (libmcrypt > 2.4.x seulement, non-libre)
- MCRYPT_PANAMA (libmcrypt > 2.4.x seulement)
- MCRYPT_RIJNDAEL_128 (libmcrypt > 2.4.x seulement)

- MCRYPT_RIJNDAEL_192 (libmcrypt > 2.4.x seulement)
- MCRYPT_RIJNDAEL_256 (libmcrypt > 2.4.x seulement)
- MCRYPT_RC2
- MCRYPT_RC4 (libmcrypt 2.2.x seulement)
- MCRYPT_RC6 (libmcrypt > 2.4.x seulement)
- MCRYPT_RC6_128 (libmcrypt 2.2.x seulement)
- MCRYPT_RC6_192 (libmcrypt 2.2.x seulement)
- MCRYPT_RC6_256 (libmcrypt 2.2.x seulement)
- MCRYPT_SAFER64
- MCRYPT_SAFER128
- MCRYPT_SAFERPLUS (libmcrypt > 2.4.x seulement)
- MCRYPT_SERPENT (libmcrypt > 2.4.x seulement)
- MCRYPT_SERPENT_128 (libmcrypt 2.2.x seulement)
- MCRYPT_SERPENT_192 (libmcrypt 2.2.x seulement)
- MCRYPT_SERPENT_256 (libmcrypt 2.2.x seulement)
- MCRYPT_SKIPJACK (libmcrypt > 2.4.x seulement)
- MCRYPT_TEAN (libmcrypt 2.2.x seulement)
- MCRYPT_THREEWAY
- MCRYPT_TRIPLEDES (libmcrypt > 2.4.x seulement)
- MCRYPT_TWOFISH (pour les vieilles versions mcrypt 2.x, ou mcrypt > 2.4.x)
- MCRYPT_TWOFISH128 (les TWOFISH xxx sont disponibles dans les nouvelles versions 2.x, mais pas dans les versions 2.4.x)
- MCRYPT_TWOFISH192
- MCRYPT_TWOFISH256
- MCRYPT_WAKE (libmcrypt > 2.4.x seulement)
- MCRYPT_XTEA (libmcrypt > 2.4.x seulement)

Vous devez (mode OFB et OFB) ou pouvez (mode CBC) fournir un vecteur d'initialisation (IV) pour ces modes de chiffrement. IV doit être unique, et avoir la même valeur au chiffrement et au déchiffrement. Pour des données qui seront enregistrées après chiffrement, vous pouvez prendre le résultat d'une fonction telle que MD5, appliquée sur le nom du fichier. Sinon, vous pouvez envoyer IV avec les données chiffrées, (reportez-vous au chapitre 9.3 de Applied Cryptography by Schneier (ISBN 0-471-11709-9) de Schneier (ISBN 0-471-11709-9) pour plus de détails sur le sujet).

8.69.8 Exemples

Mcrypt permet de chiffrer et de déchiffrer en utilisant les méthodes mentionnées ci-dessus. Les 4 commandes importantes `mcrypt_cfb` , `mcrypt_cbc` , `mcrypt_ecb` et `mcrypt_ofb`) peuvent toutes opérer en mode MCRYPT_ENCRYPT et MCRYPT_DECRYPT .

Chiffre une valeur avec un TripleDES, en mode ECB.

```
<?php
$key = "Cette clé est ultra-secrète";
$input = "Rencontrons-nous dans notre place secrète à 9 h 00.";
$encrypted_data = mcrypt_ecb(MCRYPT_TripleDES, $key, $input, MCRYPT_ENCRYPT);
?>
```

Cet exemple va retourner les données cryptées dans la variable `$encrypted_data`.

Si vous avez compilé PHP avec libmcrypt 2.4.x, ces fonctions sont toujours disponibles, mais il est vivement conseillé d'utiliser les nouvelles fonctions avancées.

Cryptage d'une valeur avec TripleDES sous 2.4.x en mode ECB

```

<?php

$key = "Ceci est une vraie clé secrète";
$input = "Rendez-vous à 9 heures, dans notre planque.";

$td = mcrypt_module_open(MCRYPT_TripleDES, "", MCRYPT_MODE_ECB, "");
$iv = mcrypt_create_iv(mcrypt_enc_get_iv_size($td), MCRYPT_RAND);

mcrypt_generic_init($td, $key, $iv);
$encrypted_data = mcrypt_generic($td, $input);
mcrypt_generic_end($td);

?>

```

Cet exemple va retourner les données cryptées dans la variable `$encrypted_data` . Pour un exemple complet, voyez [mcrypt_module_open](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Mcrypt ciphers](#)
- [Exemples](#)
- [mcrypt_cbc](#)
- [mcrypt_cfb](#)
- [mcrypt_create_iv](#)
- [mcrypt_decrypt](#)
- [mcrypt_ecb](#)
- [mcrypt_enc_get_algorithms_name](#)
- [mcrypt_enc_get_block_size](#)
- [mcrypt_enc_get_iv_size](#)
- [mcrypt_enc_get_key_size](#)
- [mcrypt_enc_get_modes_name](#)
- [mcrypt_enc_get_supported_key_sizes](#)
- [mcrypt_enc_is_block_algorithm_mode](#)
- [mcrypt_enc_is_block_algorithm](#)
- [mcrypt_enc_is_block_mode](#)
- [mcrypt_enc_self_test](#)
- [mcrypt_encrypt](#)
- [mcrypt_generic_deinit](#)
- [mcrypt_generic_end](#)
- [mcrypt_generic_init](#)
- [mcrypt_generic](#)
- [mcrypt_get_block_size](#)
- [mcrypt_get_cipher_name](#)
- [mcrypt_get_iv_size](#)
- [mcrypt_get_key_size](#)
- [mcrypt_list_algorithms](#)
- [mcrypt_list_modes](#)
- [mcrypt_module_close](#)
- [mcrypt_module_get_algo_block_size](#)
- [mcrypt_module_get_algo_key_size](#)
- [mcrypt_module_get_supported_key_sizes](#)
- [mcrypt_module_is_block_algorithm_mode](#)
- [mcrypt_module_is_block_algorithm](#)

- [mdecrypt_module_is_block_mode](#)
- [mdecrypt_module_open](#)
- [mdecrypt_module_self_test](#)
- [mdecrypt_ofb](#)
- [mdecrypt_generic](#)

8.69.10 [mdecrypt_cfb\(\)](#) : Chiffre/déchiffre des données en mode CFB

string **mdecrypt_cfb** (int cipher , string key , string data , int mode , string iv) string
mdecrypt_cfb (string cipher , string key , string data , int mode , *string iv*)

La première syntaxe utilise libmcrypt 2.2.x, et la seconde libmcrypt 2.4.x. Le paramètre mode doit être MCRYPT_ENCRYPT ou MCRYPT_DECRYPT .

[mdecrypt_cfb](#) ne doit plus être utilisée. Vous pouvez la remplacer par [mdecrypt_generic](#) et [mdecrypt_generic](#) .

8.69.11 [mdecrypt_create_iv\(\)](#) : Crée un vecteur d'initialisation à partir d'une source aléatoire

string **mdecrypt_create_iv** (int size , int source)

[mdecrypt_create_iv](#) sert à créer un IV (vecteur d'initialisation).

Le paramètre size détermine la taille de IV, le paramètre source (par défaut, un nombre aléatoire) spécifie la source de IV.

La source peut être MCRYPT RAND (générateur de nombres aléatoires système), MCRYPT_DEV_RANDOM (lecture des données depuis le fichier /dev/random) et MCRYPT_DEV_URANDOM (lecture des données depuis le fichier /dev/urandom). MCRYPT RAND est la seule à être supportée sous Windows car Windows (bien sûr) n'a pas de fichier du type /dev/random ou du type /dev/urandom .

Note

Lorsque vous utilisez MCRYPT RAND , rappelez-vous d'appeler la fonction [srand](#) avant la fonction [mdecrypt_create_iv](#) pour initialiser le générateur de nombres aléatoires ; ce n'est pas effectué automatiquement comme c'est le cas pour la fonction [rand](#) .

Exemple avec [mdecrypt_create_iv](#)

```
<?php
$size = mdecrypt_get_iv_size(MCRYPT_CAST_256, MCRYPT_MODE_CFB);
$iv = mdecrypt_create_iv($size, MCRYPT_DEV_RANDOM);
?>
```

Le VI est uniquement une solution alternative pour initialiser les routines de chiffrement. Le VI n'a pas besoin d'être spécifiquement secret, même si c'est mieux. Vous pouvez même l'envoyer avec vos données chiffrées sans perdre de sécurité.

Plus d'informations sont disponibles sur les sites de <http://www.ciphersbyritter.com/GLOSSARY.HTM#IV> , <http://fn2.freenet.edmonton.ab.ca/~jsavard/crypto/co0409.htm> et dans le chapitre 9.3 du livre "Applied Cryptography" de Schneier (ISBN 0-471-11709-9).

8.69.12 `mdecrypt_decrypt()` : Déchiffre un texte avec les paramètres donnés

`string mdecrypt_decrypt (string cipher , string key , string data , string mode , string iv)`

`mdecrypt_decrypt` déchiffre les données `data` et les retourne.

`cipher` est une constante `MCRYPT_ciphername` qui indique le nom de l'algorithme sous forme de chaîne.

`key` est la clé utilisée pour chiffrer les données. Si elle est plus petite que nécessaire, elle sera complétée avec des `'\0'`.

`data` sont les données qui doivent être déchiffrées. Si la taille des données n'est pas de la forme `n * taille_de_bloc`, elles seront complétées avec des `'\0'`. La valeur retournée peut être plus grande que la valeur d'origine.

`mode` est une constante `MCRYPT_MODE_modename` qui peut valoir : `"ecb"`, `"cbc"`, `"cfb"`, `"ofb"`, `"nofb"` ou `"stream"`.

`iv` (Vecteur d'Initialisation) est utilisé pour les modes CBC, CFB, OFB, et dans certains algorithmes de mode STREAM. Si vous ne fournissez pas le VI, alors qu'il est nécessaire, la fonction affichera une alerte, et utilisera un VI composé de caractères `'\0'`.

8.69.13 `mdecrypt_ecb()` : Obsolète : Chiffre/déchiffre des données en mode ECB

`string mdecrypt_ecb (int cipher , string key , string data , int mode)` `string mdecrypt_ecb (string cipher , string key , string data , int mode , string iv)`

La première syntaxe utilise `libmcrypt 2.2.x`, et la seconde `libmcrypt 2.4.x`. Le paramètre `mode` doit être `MCRYPT_ENCRYPT` ou `MCRYPT_DECRYPT`.

`mdecrypt_ecb` ne doit plus être utilisée. Vous pouvez la remplacer par `mdecrypt_generic` et `mdecrypt_generic`.

8.69.14 `mdecrypt_enc_get_algorithms_name()` : Retourne le nom de l'algorithme

`string mdecrypt_enc_get_algorithms_name (resource td)`

`mdecrypt_enc_get_algorithms_name` retourne le nom de l'algorithme désigné par `td`.

Exemple avec `mdecrypt_enc_get_algorithms_name`

```
<?php
$td = mdecrypt_module_open (MCRYPT_CAST_256, '', MCRYPT_MODE_CFB, '');
echo mdecrypt_enc_get_algorithms_name($td). "\n";

$td = mdecrypt_module_open ('cast-256', '', MCRYPT_MODE_CFB, '');
echo mdecrypt_enc_get_algorithms_name($td). "\n";
?>
```

L'exemple ci-dessus va afficher :

```
CAST-256
CAST-256
```

8.69.15 `mdecrypt_enc_get_block_size()` : Retourne la taille de blocs d'un algorithme

```
int mdecrypt_enc_get_block_size ( resource td )
```

`mdecrypt_enc_get_block_size` retourne la taille de blocs d'un algorithme, en octets.

8.69.16 `mdecrypt_enc_get_iv_size()` : Retourne la taille du VI d'un algorithme

```
int mdecrypt_enc_get_iv_size ( resource td )
```

`mdecrypt_enc_get_iv_size` retourne la taille du VI de l'algorithme désigné par `td`, en octets. Si la valeur retournée est 0, c'est que l'algorithme ne demande pas de VI. Un VI est demandé en mode cbc, cfb et ofb, et parfois en mode flux.

8.69.17 `mdecrypt_enc_get_key_size()` : Retourne la taille maximale de la clé pour un mode

```
int mdecrypt_enc_get_key_size ( resource td )
```

`mdecrypt_enc_get_key_size` retourne la taille maximale de clé acceptée par le mode désigné par `td`, en octets.

8.69.18 `mdecrypt_enc_get_modes_name()` : Retourne le nom du mode

```
string mdecrypt_enc_get_modes_name ( resource td )
```

`mdecrypt_enc_get_modes_name` retourne le nom du mode désigné par `td`.

Exemple avec `mdecrypt_enc_get_modes_name`

```
<?php
$td = mdecrypt_module_open(MCRYPT_CAST_256, '', MCRYPT_MODE_CFB, '');
echo mdecrypt_enc_get_modes_name($td). "\n";

$td = mdecrypt_module_open('cast-256', '', 'ecb', '');
echo mdecrypt_enc_get_modes_name($td). "\n";
?>
```

L'exemple ci-dessus va afficher :

```
CFB
```

8.69.19 `mcrypt_enc_get_supported_key_sizes()` : Retourne un tableau contenant les tailles de clés acceptées par un algorithme

`array mcrypt_enc_get_supported_key_sizes (resource td)`

`mcrypt_enc_get_supported_key_sizes` retourne un tableau contenant les tailles des clés supportées par l'algorithme désigné par `td` . S'il retourne un tableau vide, c'est que toutes les clés entre 1 et `mcrypt_enc_get_key_size` sont acceptées par l'algorithme.

Exemple avec `mcrypt_enc_get_supported_key_sizes`

```
<?php
    $td = mcrypt_module_open ('rijndael-256', '', 'ecb', '');
    var_dump (mcrypt_enc_get_supported_key_sizes($td));
?>
```

L'exemple ci-dessus va afficher :

```
array(3) {
[0]=>
int(16)
[1]=>
int(24)
[2]=>
int(32)
}
```

8.69.20 `mcrypt_enc_is_block_algorithm_mode()` : Teste le chiffrement par blocs d'un mode

`bool mcrypt_enc_is_block_algorithm_mode (resource td)`

`mcrypt_enc_is_block_algorithm_mode` retourne TRUE si ce mode utilise des algorithmes par blocs, et FALSE sinon. (i.e. FALSE pour stream, et TRUE pour cbc, cfb, ofb).

8.69.21 `mcrypt_enc_is_block_algorithm()` : Teste le chiffrement par blocs d'un algorithme

`bool mcrypt_enc_is_block_algorithm (resource td)`

`mcrypt_enc_is_block_algorithm` retourne TRUE si l'algorithme utilisé est un algorithme par blocs, et FALSE si c'est un algorithme par flux.

8.69.22 `mcrypt_enc_is_block_mode()` : Teste si le mode retourne les données par blocs

`bool mcrypt_enc_is_block_mode (resource td)`

`mcrypt_enc_is_block_mode` retourne TRUE si le mode retourne des blocs d'octets, ou bien FALSE s'il retourne des octets (par flux). (i.e. TRUE pour cbc et ecb, et FALSE pour cfb et stream).

8.69.23 `mcrypt_enc_self_test()` : Teste un module ouvert

int `mcrypt_enc_self_test` (resource td)

`mcrypt_enc_self_test` effectue un test du module ouvert et désigné par td . Si le test est concluant, elle retourne FALSE , sinon TRUE .

8.69.24 `mcrypt_encrypt()` : Chiffre un texte

string `mcrypt_encrypt` (string cipher , string key , string data , string mode , string iv)

`mcrypt_encrypt` chiffre les données, et retourne les données cryptées.

cipher est une constante MCRYPT_ciphername qui indique le nom de l'algorithme sous forme de chaîne.

key est la clé utilisée pour chiffrer les données. Si elle est plus petite que nécessaire, elle sera complétée avec des '\0'.

data sont les données qui doivent être encryptées. Si la taille des données n'est pas de la forme n * taille_de_bloc, elles seront complétées avec des '\0'. La valeur retournée peut être plus grande que la valeur d'origine.

mode est une constante MCRYPT_MODE_modename qui peut valoir : "ecb", "cbc", "cfb", "ofb", "nofb" ou "stream".

iv (Vecteur d'Initialisation) est utilisé pour les modes CBC, CFB, OFB, et dans certains algorithmes de mode STREAM. Si vous ne fournissez pas le VI, alors qu'il est nécessaire, la fonction affichera une alerte, et utilisera un vecteur d'initialisation composé de caractères '\0'.

Exemple avec `mcrypt_encrypt`

```
<?php
$iv = mcrypt_create_iv(mcrypt_get_iv_size(MCRYPT_RIJNDAEL_256, MCRYPT_MODE_ECB), MCRYPT_RAND);
$key = 'Ceci est une clé secrète';

$text = 'Rencontrons-nous à 11 heures, derrière le monument';
echo strlen($text) . "\n";

$crypttext = mcrypt_encrypt(MCRYPT_RIJNDAEL_256, $key, $text, MCRYPT_MODE_ECB, $iv);
echo strlen($crypttext) . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
42
64
```

Voir aussi `mcrypt_module_open` pour une meilleure API et un exemple.

8.69.25 `mcrypt_generic_deinit()` : Prépare le module pour le déchargement

`bool mcrypt_generic_deinit (resource td)`

`mcrypt_generic_deinit` prépare le module de chiffrement `td` pour le déchargement. Tous les tampons sont vidés, mais le module n'est pas déchargé. Vous devez appeler `mcrypt_module_close` vous-même (mais PHP le fera pour vous à la fin du script). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Pour un exemple, voir `mcrypt_module_open` et l'entrée sur `mcrypt_generic_init`.

8.69.26 `mcrypt_generic_end()` : Termine un chiffrement

`bool mcrypt_generic_end (resource td)`

Attention

Cette fonction est obsolète, utilisez `mcrypt_generic_deinit` à la place. Elle peut causer des crashes lorsqu'elle est utilisée avec `mcrypt_module_close` à cause d'une multitude de tampons libres.

`mcrypt_generic_end` termine le chiffrement désigné par le pointeur `td`. En fait, elle supprime tous les buffers, et ferme les modules utilisés. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.69.27 `mcrypt_generic_init()` : Initialise tous les buffers nécessaires

`int mcrypt_generic_init (resource td , string key , string iv)`

La taille maximale de la clé doit être celle retournée par `mcrypt_enc_get_key_size` et toutes les valeurs inférieures seront aussi valides. Le vecteur d'initialisation (VI) doit avoir la taille d'un bloc, mais vous devez lire sa taille en appelant `mcrypt_enc_get_iv_size`. IV est ignoré en mode ECB. IV DOIT exister en modes CFB, CBC, STREAM, nOFB et OFB. Il doit être aléatoire et unique (mais pas secret). Le même VI doit être utilisé pour le chiffrement et le déchiffrement. Si vous ne voulez pas l'utiliser, remplissez-le de zéros, mais ce n'est pas recommandé.

La fonction retourne une valeur négative en cas d'erreur : -3 si la taille de la clé est incorrecte, -4 quand il y a eu un problème d'allocation de mémoire et toute autre valeur en cas d'erreur inconnue. Si une erreur survient, un warning est affiché. FALSE est retourné si des paramètres incorrects sont passés à la fonction.

Vous devez appeler `mcrypt_generic_init` avant chaque appel à `mcrypt_generic` ou `mdecrypt_generic`.

Pour un exemple, voir `mcrypt_module_open`.

8.69.28 `mcrypt_generic()` : Chiffre les données

`string mcrypt_generic (resource td , string data)`

`mcrypt_generic` chiffre les données `data`. Les données sont complétées par des "\0" pour obtenir une taille multiple de la taille d'un bloc. Elle retourne les données chiffrées. Notez que la longueur

de la chaîne retournée peut être plus longue que celle passée en argument, à cause du complément.

Si vous voulez enregistrer les données chiffrées dans une base de données assurez-vous d'enregistrer la chaîne entière retournée par cette fonction, sinon la chaîne ne sera pas déchiffrée correctement. Si votre chaîne d'origine comporte 10 caractères et que la taille d'un bloc est de 8 (utilisez `mdecrypt_enc_get_block_size` pour déterminer cette taille), vous aurez besoin d'au moins 16 caractères dans le champ de votre base de données. Notez que la chaîne retournée par `mdecrypt_generic` aura 16 caractères de long... utilisez `rtrim($str, "\0")` pour supprimer le complément.

Par exemple, si vous enregistrez les données dans une base de données MySQL, souvenez-vous que les champs de type varchar suppriment automatiquement les espaces en trop durant l'insertion. Comme les données encryptées peuvent se terminer avec un espace (ASCII 32), les données seront endommagées par cette suppression. Stockez les données dans un champ tinyblob/tinytext (ou plus grand) pour que tout fonctionne normalement.

Le gestionnaire de chiffrement td doit être initialisé avec la fonction `mdecrypt_generic_init`, avec une clé et un VI, avant d'appeler cette fonction. Lorsque le chiffrement est réalisé, vous devez libérer les buffers en appelant la fonction `mdecrypt_generic_deinit`. Voyez `mdecrypt_module_open` pour un exemple.

Voir aussi `mdecrypt_generic`, `mdecrypt_generic_init` et `mdecrypt_generic_deinit`.

8.69.29 `mdecrypt_get_block_size()` : Retourne la taille de blocs d'un chiffrement

int `mdecrypt_get_block_size` (int cipher) int `mdecrypt_get_block_size` (string cipher , string module)

`mdecrypt_get_block_size` sert à lire la taille de blocs du chiffrement cipher (en combinaison avec un mode de chiffrement).

Le premier prototype sert lorsque PHP est compilé avec la bibliothèque libmcrypt 2.2.x, le second lorsqu'il est compilé avec libmcrypt 2.4.x ou 2.5.x.

Il est mieux de se servir de la fonction `mdecrypt_enc_get_block_size` comme elle utilise la ressource retournée par `mdecrypt_module_open`.

Cet exemple montre comment utiliser cette fonction lorsque PHP est compilé avec libmcrypt 2.4.x et 2.5.x.

Exemple avec `mdecrypt_get_block_size`

```
<?php
echo mdecrypt_get_block_size ('tripleDES', 'ecb'); // 8
?>
```

Voir aussi `mdecrypt_get_key_size` et `mdecrypt_encrypt`.

8.69.30 mcrypt_get_cipher_name() : Lit le nom du chiffrement utilisé

string **mcrypt_get_cipher_name** (int cipher) string **mcrypt_get_cipher_name** (string cipher)

mcrypt_get_cipher_name retourne le nom du chiffrement utilisé.

mcrypt_get_cipher_name prend le numéro de chiffrement (avec libmcrypt 2.2.x) ou prend le nom du chiffrement (avec libmcrypt 2.4.x) comme paramètre, et retourne le nom du chiffrement, ou FALSE , s'il n'existe pas.

Exemple avec mcrypt_get_cipher_name

```
<?php
$cipher = MCRYPT_TripleDES;
echo mcrypt_get_cipher_name($cipher);
?>
```

L'exemple ci-dessus va afficher :

3DES

8.69.31 mcrypt_get_iv_size() : Retourne la taille du VI utilisé par un couple chiffrement/mode

int **mcrypt_get_iv_size** (string cipher , string mode)

mcrypt_get_iv_size retourne la taille du Vecteur d'initialisation (VI). En cas d'erreur, la fonction retourne FALSE . Si le VI est ignoré dans le couple chiffrement/mode demandé, zéro est retourné.

cipher est une constante MCRYPT_ciphername qui indique le nom de l'algorithme sous forme de chaîne.

mode est une constante MCRYPT_MODE_modename qui peut valoir : "ecb", "cbc", "cfb", "ofb", "nofb" ou "stream".

Il est plus facile de servir de la fonction mcrypt_enc_get_iv_size car elle utilise la ressource retournée par mcrypt_module_open .

Exemple avec mcrypt_get_block_size

```
<?php

$size = mcrypt_get_iv_size(MCRYPT_CAST_256, MCRYPT_MODE_CFB);
$size = mcrypt_get_iv_size('des', 'ecb');

?>
```

Voir aussi mcrypt_get_block_size , mcrypt_enc_get_iv_size et mcrypt_create_iv .

8.69.32 mcrypt_get_key_size() : Retourne la taille de la clé d'un chiffrement

int **mcrypt_get_key_size** (int cipher) int **mcrypt_get_key_size** (string cipher , string module

)

La première syntaxe utilise libmcrypt 2.2.x, et la seconde libmcrypt 2.4.x ou plus récent.

mcrypt_get_key_size sert à lire la taille de clé du chiffrement cipher .

mcrypt_get_block_size prend comme argument le chiffrement cipher et retourne une taille en octets.

Cet exemple illustre comment utiliser cette fonction avec les deux bibliothèques libmcrypt 2.4.x et 2.5.x. Il est plus facile de se servir de la fonction mcrypt_enc_get_key_size comme elle utilise la ressource retournée par mcrypt_module_open .

Exemple avec mcrypt_get_block_size

```
<?php
echo mcrypt_get_key_size ('tripleDES', 'ecb'); // 24
?>
```

Voir aussi mcrypt_get_block_size , mcrypt_end_get_key_size et mcrypt_encrypt .

8.69.33 mcrypt_list_algorithms() : Liste tous les algorithmes de chiffrement supportés

array mcrypt_list_algorithms (*string* *lib_dir*)

mcrypt_list_algorithms sert à lister tous les algorithmes de chiffrement de *lib_dir* .

mcrypt_list_algorithms prend un argument optionnel, qui spécifie le dossier qui contient tous les algorithmes. S'il est omis, la valeur de `mcrypt.algorithms_dir` dans `php.ini` est utilisée.

Exemple avec mcrypt_list_algorithms

```
<?php
$algorithms = mcrypt_list_algorithms("/usr/local/lib/libmcrypt");
foreach ($algorithms as $cipher) {
    echo $cipher."\n";
}
?>
```

L'exemple ci-dessus va afficher tous les algorithmes supportés dans le dossier "`/usr/local/lib/libmcrypt`".

8.69.34 mcrypt_list_modes() : Liste tous les modes de chiffrement supportés

array mcrypt_list_modes (*string* *lib_dir*)

mcrypt_list_algorithms sert à lister tous les modes de chiffrement de *lib_dir* .

mcrypt_list_algorithms prend un argument optionnel, qui spécifie le dossier qui contient tous les algorithmes. S'il est omis, la valeur de `mcrypt.algorithms_dir` dans `php.ini` est utilisée.

Exemple avec mcrypt_list_modes

```

<?php
$modes = mdecrypt_list_modes();
foreach ($modes as $mode) {
    echo $mode . "<br />";
}
?>

```

L'exemple ci-dessus va afficher tous les modes supportés dans le dossier par défaut. Si la directive `mdecrypt.modes_dir` n'est pas définie, le dossier par défaut de `mdecrypt` est utilisé "`/usr/local/lib/libmdecrypt`".

8.69.35 `mdecrypt_module_close()` : Décharge le module de chiffrement

`bool mdecrypt_module_close (resource td)`

`mdecrypt_module_close` décharge le module `td`.

Voir `mdecrypt_module_open`.

8.69.36 `mdecrypt_module_get_algo_block_size()` : Retourne la taille de blocs d'un algorithme

`int mdecrypt_module_get_algo_block_size (string algorithm , string lib_dir)`

`mdecrypt_module_get_algo_block_size` retourne la taille de blocs d'un algorithme, en octets. Le paramètre optionnel `lib_dir` contient le chemin jusqu'au module de l'algorithme sur le système.

8.69.37 `mdecrypt_module_get_algo_key_size()` : Retourne la taille maximale de clé

`int mdecrypt_module_get_algo_key_size (string algorithm , string lib_dir)`

`mdecrypt_module_get_algo_key_size` retourne la taille maximale de la clé supportée par l'algorithme `algorithm`. Le paramètre optionnel `lib_dir` contient le chemin jusqu'au module de l'algorithme sur le système.

8.69.38 `mdecrypt_module_get_supported_key_sizes()` : Retourne un tableau contenant les tailles de clés supportées par un algorithme de chiffrement

`array mdecrypt_module_get_supported_key_sizes (string algorithm , string lib_dir)`

`mdecrypt_module_get_supported_key_sizes` retourne un tableau contenant les tailles de clés supportées par l'algorithme de chiffrement `algorithm`. S'il retourne un tableau vide, alors toutes les clés entre 1 et `mdecrypt_module_get_algo_key_size` sont supportées par l'algorithme. Le paramètre optionnel `lib_dir` peut contenir le chemin du dossier d'installation du module, sur le système.

Voir aussi `mdecrypt_enc_get_supported_key_sizes` qui est utilisée pour ouvrir les modules de chiffrement.

8.69.39 `mcrypt_module_is_block_algorithm_mode()` : Indique si un mode fonctionne par blocs

`bool mcrypt_module_is_block_algorithm_mode (string mode , string lib_dir)`

`mcrypt_module_is_block_algorithm_mode` retourne TRUE si le mode doit être utilisé avec un algorithme par bloc, sinon retourne 0 (i.e. 0 pour stream, et 1 pour cbc, cfb, ofb). Le paramètre optionnel `lib_dir` contient le chemin jusqu'au module de l'algorithme sur le système.

8.69.40 `mcrypt_module_is_block_algorithm()` : Indique si un algorithme fonctionne par blocs

`bool mcrypt_module_is_block_algorithm (string algorithm , string lib_dir)`

`mcrypt_module_is_block_algorithm` retourne TRUE si `algorithm` est un algorithme par blocs, sinon retourne 0. Le paramètre optionnel `lib_dir` contient le chemin jusqu'au module de l'algorithme sur le système.

8.69.41 `mcrypt_module_is_block_mode()` : Indique si un mode travaille par blocs

`bool mcrypt_module_is_block_mode (string mode , string lib_dir)`

`mcrypt_module_is_block_mode` retourne TRUE si ce mode fournit des blocs d'octets ou un flux d'octets. (i.e. 1 pour cbc et ecb, et 0 pour cfb et stream). Le paramètre optionnel `lib_dir` contient le chemin jusqu'au module de l'algorithme sur le système.

8.69.42 `mcrypt_module_open()` : Ouvre le module de l'algorithme et du mode à utiliser

`resource mcrypt_module_open (string algorithm , string algorithm_directory , string mode , string mode_directory)`

`mcrypt_module_open` ouvre le module de l'algorithme et du mode à utiliser. Le nom de l'algorithme est spécifié par le paramètre `algorithm` (par exemple : "twofish"), ou bien une des constantes `MCRYPT_ciphertype`. La bibliothèque est refermée en appelant `mcrypt_module_close`, mais il n'est pas nécessaire d'appeler cette fonction si `mcrypt_generic_end` est utilisé. Normalement, `mcrypt_module_open` retourne un pointeur de cryptage, ou bien FALSE en cas d'erreur.

`algorithm_directory` et `mode_directory` servent à repérer les modules de cryptage. Si vous fournissez un nom de dossier, il sera utilisé. Si vous passez une chaîne vide (""), la valeur utilisée par `mcrypt.algorithms_dir` ou `mcrypt.modes_dir` sera celle indiquée dans les directives de configuration. Lorsque ces paramètres ne sont pas fournis les valeurs par défaut, compilées avec la bibliothèque, sont utilisées. (Généralement `/usr/local/lib/libmcrypt`.)

Exemple avec `mcrypt_module_open`

```
<?php
$td = mcrypt_module_open(MCRYPT_DES, '',
    MCRYPT_MODE_ECB, '/usr/lib/mcrypt-modes');
```

```
$td = mcrypt_module_open('rijndael-256', '', 'ofb', '');
?>
```

L'exemple ci-dessus va essayer d'ouvrir le module de chiffrement par DES, dans le dossier par défaut, et le mode EBC dans le dossier /usr/lib/mcrypt-modes.

Utilisation de `mcrypt_module_open` pour encrypter

```
<?php
/* Charge un chiffrement */
$td = mcrypt_module_open('rijndael-256', '', 'ofb', '');

/* Crée le VI et détermine la taille de la clé */
$iv = mcrypt_create_iv(mcrypt_enc_get_iv_size($td), MCRYPT_DEV_RANDOM);
$ks = mcrypt_enc_get_key_size($td);

/* Crée la clé */
$key = substr(md5('very secret key'), 0, $ks);

/* Intialise le chiffrement */
mcrypt_generic_init($td, $key, $iv);

/* Chiffre les données */
$encrypted = mcrypt_generic($td, 'This is very important data');

/* Libère le gestionnaire de chiffrement */
mcrypt_generic_deinit($td);

/* Initialise le module de chiffrement pour le déchiffrement */
mcrypt_generic_init($td, $key, $iv);

/* Déchiffre les données */
$decrypted = mdecrypt_generic($td, $encrypted);

/* Libère le gestionnaire de déchiffrement, et ferme le module */
mcrypt_generic_deinit($td);
mcrypt_module_close($td);

/* Affiche la chaîne */
echo trim($decrypted)."\n";
?>
```

La première ligne de l'exemple ci-dessus va essayer d'ouvrir le chiffrement DES, dans le dossier par défaut, et le mode EBC dans le dossier /usr/lib/mcrypt-modes . Le second exemple utilise les chaînes comme nom pour le chiffrement et le mode. Cela ne fonctionne que si l'extension est compilée avec libmcrypt 2.4.x or 2.5.x.

Voir aussi [mcrypt_module_close](#) , [mcrypt_generic](#) , [mdecrypt_generic](#) , [mcrypt_generic_init](#) et [mcrypt_generic_deinit](#) .

8.69.43 `mcrypt_module_self_test()` : Teste un mode

`bool mcrypt_module_self_test (string algorithm , string lib_dir)`

`mcrypt_module_self_test` effectue un test sur l'algorithme spécifié. Le paramètre optionnel `lib_dir` contient le chemin jusqu'au module de l'algorithme sur le système.

`mcrypt_module_self_test` retourne TRUE si le test fonctionne, et FALSE sinon.

Exemple avec `mcrypt_module_self_test`

```
<?php
var_dump(mcrypt_module_self_test(MCRYPT_RIJNDAEL_128));
var_dump(mcrypt_module_self_test(MCRYPT_BOGUS_CYPHER));
?>
```

L'exemple ci-dessus va afficher :

```
bool(true)
bool(false)
```

8.69.44 mcrypt_ofb() : Chiffre/déchiffre des données en mode OFB

```
string mcrypt_ofb ( int cipher , string key , string data , int mode , string iv ) string
mcrypt_ofb ( string cipher , string key , string data , int mode , string iv )
```

La première syntaxe utilise libmcrypt 2.2.x, et la seconde libmcrypt 2.4.x. Le paramètre mode doit être MCRYPT_ENCRYPT ou MCRYPT_DECRYPT .

mcrypt_ofb ne doit plus être utilisée. Vous pouvez la remplacer par mcrypt_generic et mdcrypt_generic .

8.69.45 mdcrypt_generic() : Déchiffre

```
string mdcrypt_generic ( resource td , string data )
```

mdcrypt_generic déchiffre les données data . Notez que la longueur de la chaîne déchiffrée peut être plus longue que la chaîne originale, car elle peut avoir été complétée par des caractères "\0 ".

Exemple avec mdcrypt_generic

```
<?php
/* Données */
$key = 'Ceci est une très longue clé de chiffrement, et même trop longue!';
$plain_text = 'Ceci sont des données importantes';

/* Ouvre le module et crée un VI */
$td = mcrypt_module_open('des', '', 'ecb', '');
$key = substr($key, 0, mcrypt_enc_get_key_size($td));
$iv_size = mcrypt_enc_get_iv_size($td);
$iv = mcrypt_create_iv($iv_size, MCRYPT_RAND);

/* Initialise le module de chiffrement */
if (mcrypt_generic_init($td, $key, $iv) != -1) {

    /* Chiffre les données */
    $c_t = mcrypt_generic($td, $plain_text);
    mcrypt_generic_deinit($td);

    /* Réinitialise les tampons pour le déchiffrement */
    mcrypt_generic_init($td, $key, $iv);
    $p_t = mdcrypt_generic($td, $c_t);

    /* Nettoye */
    mcrypt_generic_deinit($td);
    mcrypt_module_close($td);
}
```

```
if (strcmp($p_t, $plain_text, strlen($plain_text)) == 0) {  
    echo "ok\n";  
} else {  
    echo "erreur\n";  
}  
?>
```

L'exemple ci-dessus montre comment vérifier que les données avant chiffrement sont bien les mêmes que celles après chiffrement/déchiffrement. Il est très important de réinitialiser le buffer de chiffrement avec [mdecrypt_generic_init](#) avant que nous ne déchiffrions les données.

Le gestionnaire de déchiffrement doit toujours être initialisé par la fonction [mdecrypt_generic_init](#) avec une clé et un VI avant d'appeler cette fonction. Lorsque le chiffrement est fait, il faut que vous libériez les données chiffrées en appelant [mdecrypt_generic_deinit](#). Voir [mdecrypt_module_open](#) pour un exemple.

Voir aussi [mdecrypt_generic](#), [mdecrypt_generic_init](#) et [mdecrypt_generic_deinit](#).

8.70 Fonctions de paiement MCVE (Monetra)

8.70.1 Introduction

Ces fonctions font l'interface avec l'API (libmonetra, anciennement connue sous le nom de libmcve) MCVE (Monetra), et permettent l'accès direct aux fonctions MCVE/Monetra depuis les scripts PHP. MCVE/Monetra est la solution de paiement en ligne de Main Street Softworks pour Linux/Unix/MacOSX/Windows (<http://www.mainstreetsoftworks.com/>) qui permet de traiter des cartes de crédits, de débit et cartes cadeaux. Il vous permet d'utiliser directement le numéro de carte de crédit, et de l'envoyer à la maison de paiement via votre serveur Unix, avec connexion modem ou internet : il permet de vous passer de services comme Authorize.Net ou Pay Flow Pro). En utilisant le module MCVE/Monetra de PHP, vous pouvez traiter des cartes de crédit directement depuis vos scripts PHP. La référence suivante va vous décrire le processus.

Note
MCVE/Monetra est la solution de remplacement pour le module de RedHat C CVS. Ils ont conclu un accord avec RedHat à la fin de l'année 2001 pour migrer tous leurs clients vers la plateforme MCVE.
Note
Cette extension a été déplacée dans le module <u>PECL</u> et ne sera plus intégrée dans PHP à partir de PHP 5.1.0.
Note
Cette extension n'est pas disponible sur les plates-formes Windows.

8.70.2 Installation

Pour activer le support MCVE (Monetra) de PHP commencez par vérifier si la bibliothèque LibMonetra (anciennement libmcve) est installée. Si vous compilez le support MCVE/Monetra directement dans PHP, vous devrez configurer PHP avec l'option `--with-mcve`. Si vous utilisez cette option sans spécifier le chemin d'accès de votre installation de libmonetra, PHP tentera de rechercher dans l'emplacement par défaut de l'installation de LibMonetra (`/usr/local`). Si Monetra (MCVE) est dans un emplacement non standard, exécutez la configuration avec : `--with-mcve=$mcve_path`, où `$mcve_path` est le chemin menant à votre installation de MCVE/Monetra. Veuillez noter que le support MCVE/Monetra requiert que `$mcve_path/lib` et `$mcve_path/include` existent et incluent `mcve.h` ou `monetra.h` dans le répertoire `include` et `libmcve.so` et/ou `libmcve.a` et/ou `libmonetra.a` dans le répertoire `lib`.

Si vous voulez installer le support MCVE/Monetra en tant que module, vous pouvez le faire en utilisant les bibliothèques PECL et taper la commande `pecl install mcve` si vous utilisez une version de PEAR au moins de 1.4.0.

Comme MCVE/Monetra vient avec une véritable séparation client/serveur, il n'y a pas de pré-requis supplémentaires pour compiler PHP avec le support MCVE. Pour tester votre extension MCVE/Monetra avec PHP, connectez-vous au site [testbox.monetra.com](http://www.testbox.monetra.com) sur le port 8333 pour le mode IP, ou le port 8444 pour le mode SSL en utilisant l'API PHP MCVE/Monetra. Utilisez 'vitale' comme nom d'utilisateur et 'test' comme mot de passe. Plus de détails sur ce système de test sont disponibles sur <http://www.mainstreetsoftworks.com/>.

8.70.3 Voir aussi

Plus de documentation sur l'API PHP de MCVE/Monetra est disponible à <http://www.mainstreetsoftworks.com/documentation.html> . La documentation éditée par Main Street est complète et est la source primaire d'informations.

8.70.4 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

M_PENDING (entier)
M_DONE (entier)
M_ERROR (entier)
M_FAIL (entier)
M_SUCCESS (entier)

- [Introduction](#)
- [Installation](#)
- [Voir aussi](#)
- [Constantes pré-définies](#)
- [m_checkstatus](#)
- [m_completeauthorizations](#)
- [m_connect](#)
- [m_connectionerror](#)
- [m_deletetrans](#)
- [m_destroyconn](#)
- [m_destroyengine](#)
- [m_getcell](#)
- [m_getcellbynum](#)
- [m_getcommadelimited](#)
- [m_getheader](#)
- [m_initconn](#)
- [m_initengine](#)
- [m_iscommadelimited](#)
- [m_maxconntimeout](#)
- [m_monitor](#)
- [m_numcolumns](#)
- [m_numrows](#)
- [m_parsecommadelimited](#)
- [m_responsekeys](#)
- [m_responseparam](#)
- [m_returnstatus](#)
- [m_setblocking](#)
- [m_setdropfile](#)
- [m_setip](#)
- [m_setssl_cafile](#)
- [m_setssl_files](#)
- [m_setssl](#)
- [m_settimeout](#)
- [m_sslcert_gen_hash](#)
- [m_transactionssent](#)
- [m_transinqueue](#)

- [m_transkeyval](#)
- [m_transnew](#)
- [m_transsend](#)
- [m_uwait](#)
- [m_validateidentifier](#)
- [m_verifyconnection](#)
- [m_verifysslcert](#)

8.70.6 **m_completeauthorizations()** : Nombre d'autorisations complètes en attente, retourne un tableau de leur identifiant

int **m_completeauthorizations** (resource conn , int array)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.7 **m_connect()** : Établit une connexion à MCVE

int **m_connect** (resource conn)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.8 **m_connectionerror()** : Obtient une représentation textuelle de la cause de l'échec de connexion

string **m_connectionerror** (resource conn)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.9 **m_deletetrans()** : Supprime une transaction spécifiée de la structure MCVE_CONN

bool **m_deletetrans** (resource conn , int identifier)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.10 **m_destroyconn()** : Détruit la connexion et la structure MCVE_CONN

bool **m_destroyconn** (resource conn)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.11 `m_destroyengine()` : Libère la mémoire associée avec la connectivité IP/SSL

void `m_destroyengine` (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.12 `m_getcell()` : Récupère une cellule spécifique d'une réponse délimitée par virgule avec le nom de colonne

string `m_getcell` (resource conn , int identifiant , string column , int row)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.13 `m_getcellbynum()` : Récupère une cellule spécifique d'une réponse délimitée par virgule avec le numéro de colonne

string `m_getcellbynum` (resource conn , int identifiant , int column , int row)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.14 `m_getcommadelimited()` : Récupère les données brutes délimitées par virgule retournées par MCVE

string `m_getcommadelimited` (resource conn , int identifiant)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.15 `m_getheader()` : Récupère le nom de la colonne dans la réponse délimitée par virgule

string `m_getheader` (resource conn , int identifiant , int column_num)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.16 `m_initconn()` : Crée et initialise une structure MCVE_CONN

resource `m_initconn` (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.17 `m_initengine()` : Apprête le client pour Communication IP/SSL

int `m_initengine` (string location)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.18 `m_iscommadelimited()` : Vérifie si la réponse est délimitée par virgule

int `m_iscommadelimited` (resource conn , int identifiant)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.19 `m_maxconntimeout()` : Le temps maximal que l'API tentera une connexion à MCVE

bool `m_maxconntimeout` (resource conn , int secs)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.20 `m_monitor()` : Effectue une communication avec MCVE (envoi/réception de données) en mode non-bloquant

int `m_monitor` (resource conn)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.21 `m_numcolumns()` : Nombre de colonnes retournées dans une réponse délimitée par virgule

int `m_numcolumns` (resource conn , int identifiant)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.22 `m_numrows()` : Nombre de lignes retournés dans une réponse délimitée par virgule

int `m_numrows` (resource conn , int identifiant)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.23 `m_parsecommadelimited()` : Analyse la réponse délimitée par virgule alors `m_getcell`, etc. fonctionnera

int `m_parsecommadelimited` (resource conn , int identifieur)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.24 `m_responsekeys()` : Retourne un tableau de chaînes de caractères qui représente les clés qui peuvent être utilisées en tant que paramètres de réponse sur cette transaction

array `m_responsekeys` (resource conn , int identifieur)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.25 `m_responseparam()` : Récupère un paramètre spécial de réponse

string `m_responseparam` (resource conn , int identifieur , string key)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.26 `m_returnstatus()` : Vérifie si la transaction fut complétée correctement

int `m_returnstatus` (resource conn , int identifieur)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.27 `m_setblocking()` : Fixe le mode bloquant/non-bloquant pour une connexion

int `m_setblocking` (resource conn , int tf)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.28 `m_setdropfile()` : Fixe la méthode de connexion à Drop-File

int `m_setdropfile` (resource conn , string directory)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.29 **m_setip()** : Fixe la méthode de connexion à IP

int **m_setip** (resource conn , string host , int port)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.30 **m_setssl_cafile()** : Fixe le fichier SSL CA (Certificate Authority) pour vérification du certificat du serveur

int **m_setssl_cafile** (resource conn , string cafile)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.31 **m_setssl_files()** : Fixe les fichiers de clé de certificat et certifie si le serveur requiert la vérification du client par certificat

int **m_setssl_files** (resource conn , string sslkeyfile , string sslcertfile)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.32 **m_setssl()** : Fixe la méthode de connexion à SSL

int **m_setssl** (resource conn , string host , int port)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.33 **m_settimeout()** : Fixe le temps maximal d'une transaction (par transaction)

int **m_settimeout** (resource conn , int seconds)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.34 **m_sslcert_gen_hash()** : Génère une clé de hachage pour les vérifications de certificat de client SSL

string **m_sslcert_gen_hash** (string filename)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.35 `m_transactionssent()` : Vérifie si le tampon de sortie est vide

int `m_transactionssent` (resource conn)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.36 `m_transinqueue()` : Nombre de transactions dans la file de client

int `m_transinqueue` (resource conn)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.37 `m_transkeyval()` : Ajoute une paire clé/valeur d'une transaction. Remplace `transparam()` dépréciée

int `m_transkeyval` (resource conn , int identifier , string key , string value)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.38 `m_transnew()` : Démarre une nouvelle transaction

int `m_transnew` (resource conn)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.39 `m_transsend()` : Finalise et envoie la transaction

int `m_transsend` (resource conn , int identifier)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.40 `m_uwait()` : Attend x micro secondes

int `m_uwait` (int microsecs)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.41 `m_validateidentifrier()` : Valide ou non l'identifiant passé sur n'importe quelle transaction qu'il est passé

int `m_validateidentifrier` (resource conn , int tf)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.42 **m_verifyconnection()** : Ping ou non sur la connexion pour la vérifier

bool **m_verifyconnection** (resource conn , int tf)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.70.43 **m_verifysslcert()** : Vérifie ou non le certificat du serveur SSL

bool **m_verifysslcert** (resource conn , int tf)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.71 Fonctions Memcache

8.71.1 Introduction

Le module Memcache fournit une interface procédurale maniable ainsi qu'orientée objet à Memcache , un démon fortement efficace dans la gestion du cache, qui est principalement destiné à faire baisser la charge des bases de données dans les applications web dynamiques.

Ce module n'a pas un support natif des serveurs multiples, mais vous pouvez tout de même l'implémenter vous-même dans vos applications. Établir plusieurs connexions memcache, définir les priorités pour chaque serveur, etc.

Plus d'informations concernant Memcache peuvent être consultées sur <http://www.danga.com/memcached/> .

8.71.2 Pré-requis

Ce module utilise les fonctions de la bibliothèque [zlib](#) pour supporter la compression des données à la volée. Zlib est donc requis pour installer ce module.

PHP 4.3.3 ou supérieur est requis pour utiliser l'extension Memcache .

8.71.3 Installation

Cette extension [PECL](#) n'est pas intégrée à PHP. Des informations comme les notes sur les nouvelles versions, les téléchargements, les sources des fichiers, les informations concernant les mainteneurs ainsi qu'un CHANGELOG, peuvent être trouvées ici : <http://pecl.php.net/package/memcache> .

Pour utiliser ces fonctions, vous devez compiler PHP avec le support MemCache en utilisant l'option de configuration `--enable-memcache[=DIR]` .

Les utilisateurs de Windows doivent activer la bibliothèque `php_memcache.dll` dans le `php.ini` pour pouvoir utiliser ces fonctions. Vous pouvez télécharger ces bibliothèques DLL pour ces extensions PECL depuis la page [de téléchargement de PHP](#) sous sur la page <http://snaps.php.net/> .

8.71.4 Constantes pré-définies

Nom	Description
MEMCACHE_COMPRESSED (entier)	Utilisée pour activer à la volée la compression des données avec les fonctions <code>Memcache::set</code> , <code>Memcache::add</code> et <code>Memcache::replace</code> .

8.71.5 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.71.6 Types de ressources

Il n'y a qu'une seule ressource utilisée par le module Memcache : c'est le lien identifiant représentant la connexion du serveur de cache.

8.71.7 Exemples

Exemple avec l'extension Memcache

```
<?php

$memcache = new Memcache;
$memcache->connect('localhost', 11211) or die ("Connexion impossible");

$version = $memcache->getVersion();
echo "Version du serveur : ".$version."<br />\n";

$tmp_object = new stdClass;
$tmp_object->str_attr = 'test';
$tmp_object->int_attr = 123;

$memcache->set('key', $tmp_object, false, 10) or die ("Echec de la sauvegarde des données sur le cache");
echo "Les données ont été stockées dans le cache (les données expireront dans 10 secondes)<br />\n";

$get_result = $memcache->get('key');
echo "Données depuis le cache :<br />\n";

var_dump($get_result);

?>
```

Dans l'exemple ci-dessus, un objet est sauvegardé dans le cache et ensuite, affiché. Les objets ainsi que les autres types non-scalaires sont linéarisés avant d'être sauvegardé ; il est donc impossible d'y stocker des ressources (i.e. identifiant de connexion et autres) dans le cache.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Constantes pré-définies](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Exemples](#)
- [Memcache::add](#)
- [Memcache::close](#)
- [Memcache::connect](#)
- [memcache_debug](#)
- [Memcache::decrement](#)
- [Memcache::delete](#)
- [Memcache::flush](#)
- [Memcache::get](#)
- [Memcache::getStats](#)
- [Memcache::getVersion](#)
- [Memcache::increment](#)
- [Memcache::pconnect](#)
- [Memcache::replace](#)
- [Memcache::set](#)

8.71.9 Memcache::close() : Ferme la connexion avec le serveur Memcache

bool **Memcache::close** (void)

Memcache::close ferme la connexion au serveur Memcache . Cette fonction ne ferme pas les connexions persistantes qui seront fermées uniquement lors du redémarrage du serveur web.

Vous pouvez également utiliser la fonction memcache_close . Regardez l'exemple ci-dessous.

Exemple avec Memcache::close

```
<?php

/* API procédurale */
$memcache_obj = memcache_connect('memcache_host', 11211);
/*
faites quelque chose ici ..
*/
memcache_close($memcache_obj);

/* API orientée objet */
$memcache_obj = new Memcache;
$memcache_obj->connect('memcache_host', 11211);
/*
faites quelque chose ici ..
*/
$memcache_obj->close();

?>
```

Memcache::close retourne FALSE si une erreur survient.

Voir aussi Memcache::connect et Memcache::pconnect .

8.71.10 Memcache::connect() : Ouvre une connexion avec le serveur Memcache

bool **Memcache::connect** (string host , int port , int timeout)

Memcache::connect établie une connexion avec le serveur de cache Memcache . Les paramètres host et port désignent respectivement l'hôte et le port où le serveur Memcache attend des connexions. Le paramètre port est optionnel, la valeur par défaut est 11211. Vous pouvez également définir un paramètre timeout (en secondes), qui sera utiliser lors de la connexion avec les serveur. Faites attention si vous changez la valeur par défaut : vous pouvez perdre tous les avantages du système de cache si votre connexion est trop lente.

La connexion, qui a été ouverte en utilisant la fonction Memcache::connect sera automatiquement fermée à la fin de votre script. Vous pouvez néanmoins la refermer en utilisant la fonction Memcache::close .

Vous pouvez également utiliser la fonction memcache_connect . Regardez l'exemple ci-dessous.

Exemple avec Memcache::connect

```
<?php

/* API procédurale */
```

```

$memcache_obj = memcache_connect('memcache_host', 11211);

/* API orientée objet */

$memcache = new Memcache;
$memcache->connect('memcache_host', 11211);

?>

```

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [Memcache::pconnect](#) et [Memcache::close](#) .

8.71.11 memcache_debug() : Active ou non l'affichage des informations de déboguage

bool **memcache_debug** (int on_off)

[memcache_debug](#) active l'affichage des informations de déboguage si le paramètre on_off vaut 1 et le désactive s'il vaut 0.

Note

[memcache_debug](#) est accessible uniquement si PHP a été compilé avec l'option --enable-debug et retournera toujours TRUE dans ce cas. Sinon, cette fonction n'aura aucun effet et retournera toujours FALSE .

8.71.12 Memcache::decrement() : Décrémente la valeur d'un élément

int **Memcache::decrement** (string key , int value)

[Memcache::decrement](#) décrémente la valeur de l'élément par value . De la même façon que la fonction [memcache::increment](#) , la valeur courante de l'élément est tout d'abord convertie en numérique et seulement ensuite, on soustrait la valeur value .

Le paramètre value est optionnel. Par défaut, il vaut 1.

Note

La nouvelle valeur de l'élément ne peut être inférieure à zéro.

Note

N'utilisez pas la fonction [Memcache::decrement](#) avec les éléments stockés compressés. Dans ce cas, l'appel à la fonction [Memcache::get](#) échouera.

Vous pouvez également utiliser la fonction [memcache_decrement](#) . Regardez l'exemple ci-dessous.

Exemple avec [Memcache::decrement](#)

```

<?php

/* API procédurale */
$memcache_obj = memcache_connect('memcache_host', 11211);
/* decrement item by 2 */
$new_value = memcache_decrement($memcache_obj, 'test_item', 2);

/* API orientée objet */
$memcache_obj = new Memcache;
$memcache_obj->connect('memcache_host', 11211);

```

```

/* décrémentation de l'élément par 3 */
$new_value = $memcache_obj->decrement('test_item', 3);
?>

```

Memcache::decrement **ne crée pas** l'élément s'il n'existe pas.

Memcache::decrement retourne la nouvelle valeur de l'élément en cas de succès ou FALSE en cas d'échec.

Voir aussi Memcache::increment et Memcache::replace .

8.71.13 Memcache::delete() : Efface un élément du serveur de cache

bool **Memcache::delete** (string key , int timeout)

Memcache::delete efface l'élément identifié par la clé key . Si le paramètre timeout est spécifié, l'élément expirera après timeout secondes.

Vous pouvez également utiliser la fonction memcache_delete . Regardez l'exemple ci-dessous.

Exemple avec Memcache::delete

```

<?php

/* API procédurale */
$memcache_obj = memcache_connect('memcache_host', 11211);

/* après 10 secondes, l'élément sera effacé par la serveur de cache */
memcache_delete($memcache_obj, 'key_to_delete', 10);

/* API orientée objet */
$memcache_obj = new Memcache;
$memcache_obj->connect('memcache_host', 11211);

$memcache_obj->delete('key_to_delete', 10);

?>

```

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.71.14 Memcache::flush() : Efface tous les éléments existant sur le serveur de cache

bool **Memcache::flush** (void)

Memcache::flush invalide immédiatement tous les éléments existant sur le serveur de cache.

Memcache::flush ne libère aucune ressource actuellement, il marque uniquement tous les éléments comme ayant expirés, donc la mémoire occupée sera réutilisée avec de nouveaux éléments.

Vous pouvez également utiliser la fonction memcache_flush . Regardez l'exemple ci-dessous.

Exemple avec Memcache::flush

```

<?php

/* API procédurale */
$memcache_obj = memcache_connect('memcache_host', 11211);

```

```

memcache_flush($memcache_obj);

/* API orientée objet */

$memcache_obj = new Memcache;
$memcache_obj->connect('memcache_host', 11211);

$memcache_obj->flush();

?>

```

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.71.15 Memcache::get() : Récupère un élément du serveur de cache

string **Memcache::get** (string key) string **Memcache::get** (array keys)

Memcache::get retourne les données précédemment stockées dans l'élément identifié par la clé key s'il existe sur le serveur au moment de l'appel.

Exemple avec Memcache::get

```

<?php

/* API procédurale */
$memcache_obj = memcache_connect('memcache_host', 11211);
$var = memcache_get($memcache_obj, 'some_key');

/* API orientée objet */
$memcache_obj = new Memcache;
$memcache_obj->connect('memcache_host', 11211);
$var = $memcache_obj->get('some_key');

/*
Vous pouvez également utiliser un tableau de clés en tant que paramètre.
Si un tel élément n'est pas trouvé sur le serveur, le tableau
résultat ne comprendra simplement pas une telle clé.
*/

/* API procédurale */
$memcache_obj = memcache_connect('memcache_host', 11211);
$var = memcache_get($memcache_obj, Array('some_key', 'another_key'));

/* API Orientée Objet */
$memcache_obj = new Memcache;
$memcache_obj->connect('memcache_host', 11211);
$var = $memcache_obj->get(Array('some_key', 'second_key'));

?>

```

Memcache::get retourne FALSE en cas d'échec ou si la clé key n'a pas été trouvée.

8.71.16 Memcache::getStats() : Lit les statistiques du serveur

array **Memcache::getStats** (void)

Memcache::getStats retourne un tableau associatif avec les statistiques du serveur. Les index du tableau correspondent aux paramètres de statistiques, et la valeur associée est la valeur de ces statistiques.

Vous pouvez aussi utiliser la fonction [memcache_get_stats](#) .

[Memcache::getstats](#) retourne FALSE en cas d'erreur. error.

Voir aussi [Memcache::getVersion](#) .

8.71.17 Memcache::getVersion() : Retourne le numéro de version du serveur

string **Memcache::getVersion** (void)

[Memcache::getVersion](#) retourne une chaîne avec le numéro de version du serveur.

Vous pouvez aussi utiliser la fonction [memcache_get_version](#) . Voyez l'exemple ci-dessous.

Exemple avec [Memcache::getVersion](#)

```
<?php
/* API procédurale */
$memcache_obj = memcache_connect('memcache_host', 11211);

echo memcache_get_version($memcache_obj);

/* API OOP */
$memcache_obj = new Memcache;
echo $memcache_obj->getVersion();
?>
```

[Memcache::getVersion](#) retourne FALSE en cas d'erreur.

Voir aussi [Memcache::getStats](#) .

8.71.18 Memcache::increment() : Incrément la valeur d'un élément

int **Memcache::increment** (string key , int value)

[Memcache::increment](#) incrémente la valeur d'un élément identifié par la clé key par la valeur value . Si l'élément identifié par la clé key n'est pas de type numérique et ne peut être converti en nombre, la valeur de cette élément sera défini à value .

Le paramètre value est optionnel. Par défaut, il vaut 1.

Note

N'utilisez pas [memcache::increment](#) avec les éléments stockés compressés. Dans ce cas, l'appel à la fonction [Memcache::get](#) échouera.

Vous pouvez également utiliser la fonction [memcache_increment](#) . Regardez l'exemple ci-dessous.

Exemple avec [Memcache::increment](#)

```
<?php
/* API procédurale */
$memcache_obj = memcache_connect('memcache_host', 11211);
/* incrémentation du compteur de 2 */
```

```

$current_value = memcache_increment($memcache_obj, 'counter', 2);

/* API orientée objet */
$memcache_obj = new Memcache;
$memcache_obj->connect('memcache_host', 11211);
/* incrémentation du compteur de 3 */
$current_value = $memcache_obj->increment('counter', 3);

?>

```

Memcache::increment retourne la nouvelle valeur de l'élément en cas de succès ou FALSE en cas d'échec.

Memcache::increment **ne crée pas** l'élément s'il n'existe pas.

Voir aussi Memcache::decrement et Memcache::replace .

8.71.19 Memcache::pconnect() : Ouvre un connexion persistante à un serveur de cache

bool **Memcache::pconnect** (string host , int port , int timeout)

Memcache::pconnect est similaire à la fonction Memcache::connect avec la différence que la connexion sera persistante. Ce type de connexion n'est pas fermé à la fin du script et par la fonction Memcache::close .

Vous pouvez également utiliser la fonction memcache_pconnect . Regardez l'exemple ci-dessous.

Exemple avec Memcache::pconnect

```

<?php

/* API procédurale */
$memcache_obj = memcache_pconnect('memcache_host', 11211);

/* API orientée objet */

$memcache_obj = new Memcache;
$memcache_obj->pconnect('memcache_host', 11211);

?>

```

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi Memcache::connect .

8.71.20 Memcache::replace() : Remplace une valeur d'un élément existant

bool **Memcache::replace** (string key , mixed var , int flag , int expire)

Memcache::replace est utilisé pour remplacer la valeur d'un élément identifié par la clé key . Dans le cas où l'élément identifié par la clé key n'existe pas, la fonction Memcache::replace retournera FALSE . Pour le reste, la fonction Memcache::replace fonctionne de la même façon que la fonction Memcache::set .

Vous pouvez également utiliser la fonction `memcache_replace` . Regardez l'exemple ci-dessous.

Exemple avec `Memcache::replace`

```
<?php
$memcache_obj = memcache_connect('memcache_host', 11211);

/* API procédurale */
memcache_replace($memcache_obj, "test_key", "some variable", false, 30);

/* API orientée objet */
$memcache_obj->replace("test_key", "some variable", false, 30);

?>
```

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `Memcache::set` et `Memcache::add` .

8.71.21 `Memcache::set()` : Stocke des données dans le serveur de cache

bool **Memcache::set** (string key , mixed var , int flag , int expire)

`Memcache::set` stocke l'élément var avec la clé key sur le serveur de cache. Le paramètre expire représente le délai en secondes d'expiration de l'élément. S'il vaut 0, l'élément n'expirera jamais (mais le serveur de cache ne garantit pas que cet élément sera toujours stocké, il peut être effacé du cache pour faire de la place à d'autres éléments).

Vous pouvez utiliser la constante `MEMCACHE_COMPRESSED` comme valeur du paramètre flag si vous voulez utiliser la compression à la volée (utilisation de la bibliothèque zlib).

Vous pouvez également utiliser la fonction `memcache_set` . Regardez l'exemple ci-dessous.

Note

Souvenez-vous que les ressources (i.e. identifiant de fichiers ou de connexion) ne peuvent pas être stockées dans le cache, car elles ne peuvent pas être représentées linéairement.

Exemple avec `Memcache::set`

```
<?php
/* API procédurale */

/* connexion au serveur de cache */
$memcache_obj = memcache_connect('memcache_host', 11211);

/*
définie la valeur de l'élément identifié par la clé 'var_key' ;
utilisation de la valeur 0 pour le flag ;
la compression n'est pas utilisée ;
le délai d'expiration vaut 30 secondes
*/
memcache_set($memcache_obj, 'var_key', 'quelques variables', 0, 30);

echo memcache_get($memcache_obj, 'var_key');

?>
```

Exemple avec `Memcache::set`

```
<?php
```

```
/* API orientée objet */  
  
$memcache_obj = new Memcache;  
  
/* connexion au serveur de cache */  
$memcache_obj->connect('memcache_host', 11211);  
  
/*  
définie la valeur de l'élément identifié par la clé 'var_key' ;  
utilisation de la compression à la volée ;  
le délai d'expiration vaut 50 secondes  
*/  
$memcache_obj->set('var_key', 'quelques grosses variables', MEMCACHE_COMPRESSED, 50);  
  
echo $memcache_obj->get('var_key');  
  
?>
```

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [Memcache::add](#) et [Memcache::replace](#) .

8.72 Hash

8.72.1 Introduction

Ces fonctions ont été prévues pour fonctionner avec [mhash](#) . Mhash peut être utilisée pour créer une somme de contrôle, un hachage et bien plus.

Cet ensemble de fonctions représente une interface avec la bibliothèque mhash. Mhash accepte un grand nombre d'algorithmes différents, tels MD5, SHA1, GOST et bien d'autres. Pour une liste complète des hashes supportés, reportez-vous à la documentation de mhash. La règle générale est que vous pouvez accéder à un algorithme depuis PHP avec la constante MHASH_NOMDUHASH. Par exemple, pour accéder à l'algorithme TIGER, vous pouvez utiliser la constante MHASH_TIGER.

8.72.2 Pré-requis

Pour l'utiliser, téléchargez les distributions de mhash depuis ce [site internet](#) et suivez les instructions d'installation incluses.

8.72.3 Installation

Vous aurez besoin de compiler PHP avec l'option `--with-mhash[=DIR]` pour activer cette extension. DIR est le chemin du dossier d'installation de la bibliothèque MHASH.

8.72.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.72.5 Types de ressources

Cette extension ne définit aucune ressource.

8.72.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Voici une liste des modes qui sont supportés par mhash. Toutes les constantes ne sont pas ici, et vous pouvez supposer que cette documentation n'est plus à jour.

- MHASH_ADLER32
- MHASH_CRC32
- MHASH_CRC32B
- MHASH_GOST
- MHASH_HAVAL128
- MHASH_HAVAL160

- MHASH_HAVAL192
- MHASH_HAVAL256
- MHASH_MD4
- MHASH_MD5
- MHASH_RIPEMD160
- MHASH_SHA1
- MHASH_SHA256
- MHASH_TIGER
- MHASH_TIGER128
- MHASH_TIGER160

8.72.7 Exemples

Calcule le MD5 et le hmac, puis l'affiche comme un hexadécimal

```
<?php
$input = "what do ya want for nothing?";
$hash = mhash (MHASH_MD5, $input);
echo "Le hash vaut ".bin2hex ($hash)."<br />\n";
$hash = mhash (MHASH_MD5, $input, "Jefe");
echo "Le hmac vaut ".bin2hex ($hash)."<br />\n";
?>
```

L'exemple ci-dessus va afficher :

Le hmac vaut d03cb659cbf9192dcd066272249f8412
Le hmac vaut 750c783e6ab0b503eaa86e310a5db738

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [mhash_count](#)
- [mhash_get_block_size](#)
- [mhash_get_hash_name](#)
- [mhash_keygen_s2k](#)
- [mhash](#)

8.72.9 mhash_get_block_size() : Retourne la taille de bloc du hash

int [mhash_get_block_size](#) (int hash)

[mhash_get_block_size](#) sert à connaître la taille de bloc du hash spécifié hash .

[mhash_get_block_size](#) prend un seul argument : le hash et retourne la taille en octets, ou bien FALSE si le hash n'existe pas.

8.72.10 `mhash_get_hash_name()` : Retourne le nom du hash

string `mhash_get_hash_name` (int hash)

`mhash_get_hash_name` sert à connaître le nom d'un hash.

`mhash_get_hash_name` prend un numero d'identifiant de hash, et retourne son nom ou FALSE si le hash n'existe pas, ou si une erreur est survenue.

Exemple avec `mhash_get_hash_name`

```
<?php
    $hash = MHASH_MD5;
    print mhash_get_hash_name($hash);
?>
```

L'exemple ci-dessus va afficher :

MD5

8.72.11 `mhash_keygen_s2k()` : Génère une clé

string `mhash_keygen_s2k` (int hash , string password , string salt , int bytes)

`mhash_keygen_s2k` génère une clé de bytes octets de long, à partir d'un mot de passe. Cette fonction utilise l'algorithme Salted S2K, spécifié dans OpenPGP (RFC 2440). Cet algorithme va utiliser l'algorithme de hashage hash pour créer la clé. Le paramètre salt doit être différent et suffisamment aléatoire pour chaque clé que vous générez, afin d'en créer des différentes. Ce grain de sel ("salt") reservira lorsque vous vérifierez les clés : c'est alors une bonne idée que de l'ajouter à la fin de la clé générée. salt doit avoir la longueur de 8 octets, et sera complété par des 0 si vous ne fournissez par suffisamment de données.

N'oubliez pas que les mots de passe fournis par les utilisateurs ne sont pas conseillés pour faire des clés cryptographiques, étant donné que les utilisateurs normaux retiennent des mots de passe qu'ils peuvent saisir au clavier. Ces mots de passe utilisent uniquement 6 à 7 des 8 bits d'un caractère (voire moins). Il est vivement recommandé d'appliquer une fonction de transformation (comme celle-ci), à un mot de passe utilisateur.

8.72.12 `mhash()` : Calcule un hash

string `mhash` (int hash , string data , *string* key)

`mhash` applique la fonction de hash hash aux données data et retourne le résultat. Si le paramètre key est spécifié, il retournera le HMAC résultant. HMAC est un hash indexé utilisé pour l'identification de message, ou bien un simple rapport de message, suivant la clé spécifiée. Certains algorithmes supportés dans mhash ne sont pas compatibles avec le mode HMAC. En cas d'erreur, la fonction retourne FALSE .

8.73 Ming pour Flash

Attention

Ce module est **EXPERIMENTAL** . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez ce module à vos risques et périls.

8.73.1 Introduction

Avant tout, sachez que Ming n'est pas un acronyme. Ming est une bibliothèque open-source (LGPL) qui vous permet de créer des animations au format Flash. Ming supporte toutes les fonctionnalités de Flash 4 : les formes (shapes), les gradients, les images bitmaps (JPEG et PNG), les morphing (transformations d'une forme en une autre), les textes, actions, sprites (mini animations), le streaming MP3 et les transformations de couleurs. Le seul ajout futur est celui des événements sons.

Notez que toutes les distances spécifiées (longueurs, distances, tailles...) sont en "twips", c'est-à-dire 20 unités par pixels. C'est plus ou moins arbitraire, car le lecteur Flash fait une mise à l'échelle avec les valeurs qui lui sont fournies dans la balise embed, ou la frame courante si la balise embed n'est pas utilisée.

Ming propose de nombreux avantages par rapport à l'extension swf. Vous pouvez utiliser Ming sur tous les OS où vous pouvez compiler le code, tandis que swf est limité à Windows. Ming vous évite la déconcertante complexité du format SWF, en transformant les éléments des animations en objets PHP. Enfin, Ming est toujours en cours de développement et surveillé par son auteur : si vous souhaitez une nouvelle fonctionnalité, dites-le-lui : ming@opaque.net .

Ming et tous les objets cités ont été ajoutés en PHP 4.0.5.

8.73.2 Pré-requis

Pour utiliser Ming avec PHP, vous devez d'abord installer la library Ming. Le code source et les instructions d'installation sont disponibles sur la page d'accueil de Ming : <http://ming.sourceforge.net/> , avec des exemples un tutorial et l'actualité Ming.

Téléchargez l'archive Ming. Décompressez-la et allez dans le dossier Ming. Faites "make", puis "make install".

Cela va compiler le fichier libming.so et l'installer dans /usr/lib/ , et copier ming.h into /usr/include/ . Editez la ligne PREFIX= dans le fichier Makefile pour indiquer votre dossier d'installation.

8.73.3 Installation

Compilation CGI avec PHP (Unix)

```
mkdir <phpdir>/ext/ming
cp php_ext/* <phpdir>/ext/ming
cd <phpdir>
./buildconf
./configure --with-ming <other config options>
```

Compilez et installez PHP comme d'habitude. Redémarrez votre serveur web si nécessaire.

Maintenant, soit vous ajoutez simplement `extension=php_ming.so` dans votre fichier `php.ini`, soit vous ajoutez la ligne `dl('php_ming.so');` au début de tous vos scripts qui utilisent Ming.

8.73.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.73.5 Types de ressources

8.73.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

MING_NEW ([entier](#))
MING_ZLIB ([entier](#))
SWFBUTTON_HIT ([entier](#))
SWFBUTTON_DOWN ([entier](#))
SWFBUTTON_OVER ([entier](#))
SWFBUTTON_UP ([entier](#))
SWFBUTTON_MOUSEUPOUTSIDE ([entier](#))
SWFBUTTON_DRAGOVER ([entier](#))
SWFBUTTON_DRAGOUT ([entier](#))
SWFBUTTON_MOUSEUP ([entier](#))
SWFBUTTON_MOUSEDOWN ([entier](#))
SWFBUTTON_MOUSEOUT ([entier](#))
SWFBUTTON_MOUSEOVER ([entier](#))
SWFFILL_RADIAL_GRADIENT ([entier](#))
SWFFILL_LINEAR_GRADIENT ([entier](#))
SWFFILL_TILED_BITMAP ([entier](#))
SWFFILL_CLIPPED_BITMAP ([entier](#))
SWFTEXTFIELD_HASLENGTH ([entier](#))
SWFTEXTFIELD_NOEDIT ([entier](#))
SWFTEXTFIELD_PASSWORD ([entier](#))
SWFTEXTFIELD_MULTILINE ([entier](#))
SWFTEXTFIELD_WORDWRAP ([entier](#))
SWFTEXTFIELD_DRAWBOX ([entier](#))
SWFTEXTFIELD_NOSELECT ([entier](#))
SWFTEXTFIELD_HTML ([entier](#))
SWFTEXTFIELD_ALIGN_LEFT ([entier](#))
SWFTEXTFIELD_ALIGN_RIGHT ([entier](#))
SWFTEXTFIELD_ALIGN_CENTER ([entier](#))
SWFTEXTFIELD_ALIGN_JUSTIFY ([entier](#))
SWFACTION_ONLOAD ([entier](#))
SWFACTION_ENTERFRAME ([entier](#))
SWFACTION_UNLOAD ([entier](#))
SWFACTION_MOUSEMOVE ([entier](#))
SWFACTION_MOUSEDOWN ([entier](#))
SWFACTION_MOUSEUP ([entier](#))
SWFACTION_KEYDOWN ([entier](#))
SWFACTION_KEYUP ([entier](#))

SWFACTION_DATA ([entier](#))

8.73.7 Classes pré-définies

Ces classes sont définies par cette extension, et elles ne seront disponibles que si cette extension a été compilée avec PHP, ou bien chargée dynamiquement.

Ming introduit 13 objets en PHP. Pour les utiliser, vous devez être familier avec les [objets](#) .

swfshape
swffill
swfgradient
swfbitmap
swftext
swftextfield
swffont
swfdisplayitem
swfmovie
swfbutton
swfaction
swfmorph
swfsprite

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Classes pré-définies](#)
- [ming_keypress](#)
- [ming_setcubicthreshold](#)
- [ming_setscale](#)
- [ming_useConstants](#)
- [ming_useswfversion](#)
- [SWFAction](#)
- [SWFBitmap->getHeight](#)
- [SWFBitmap->getWidth](#)
- [SWFBitmap](#)
- [SWFbutton->addAction](#)
- [SWFButton::addASound](#)
- [SWFbutton->addShape](#)
- [SWFbutton->setAction](#)
- [SWFbutton->setdown](#)
- [SWFbutton->setHit](#)
- [SWFButton::setMenu](#)
- [SWFbutton->setOver](#)
- [SWFbutton->setUp](#)
- [SWFbutton](#)
- [SWFDisplayItem::addAction](#)
- [SWFDisplayItem->addColor](#)
- [SWFDisplayItem::endMask](#)
- [SWFDisplayItem::getRot](#)

- [SWFDisplayItem::getX](#)
- [SWFDisplayItem::getXScale](#)
- [SWFDisplayItem::getXSkew](#)
- [SWFDisplayItem::getY](#)
- [SWFDisplayItem::getYScale](#)
- [SWFDisplayItem::getYSkew](#)
- [SWFDisplayItem->move](#)
- [SWFDisplayItem->moveTo](#)
- [SWFDisplayItem->multColor](#)
- [SWFDisplayItem->remove](#)
- [SWFDisplayItem->Rotate](#)
- [SWFDisplayItem->rotateTo](#)
- [SWFDisplayItem->scale](#)
- [SWFDisplayItem->scaleTo](#)
- [SWFDisplayItem->setDepth](#)
- [SWFDisplayItem::setMaskLevel](#)
- [SWFDisplayItem::setMatrix](#)
- [SWFDisplayItem->setName](#)
- [SWFDisplayItem->setRatio](#)
- [SWFDisplayItem->skewX](#)
- [SWFDisplayItem->skewXTo](#)
- [SWFDisplayItem->skewY](#)
- [SWFDisplayItem->skewYTo](#)
- [SWFFill->moveTo](#)
- [SWFFill->rotateTo](#)
- [SWFFill->scaleTo](#)
- [SWFFill->skewXTo](#)
- [SWFFill->skewYTo](#)
- [SWFFill](#)
- [SWFFont::getAscent](#)
- [SWFFont::getDescent](#)
- [SWFFont::getLeading](#)
- [SWFFont::getShape](#)
- [SWFFont::getUTF8Width](#)
- [swffont->getwidth](#)
- [SWFFont](#)
- [SWFFontChar::addChars](#)
- [SWFFontChar::addUTF8Chars](#)
- [SWFGradient->addEntry](#)
- [SWFGradient](#)
- [SWFMorph->getshape1](#)
- [SWFMorph->getshape2](#)
- [SWFMorph](#)
- [SWFMovie->add](#)
- [SWFMovie::addExport](#)
- [SWFMovie::addFont](#)
- [SWFMovie::importChar](#)
- [SWFMovie::importFont](#)
- [SWFMovie::labelFrame](#)
- [SWFMovie->nextframe](#)
- [SWFMovie->output](#)
- [swfmovie->remove](#)
- [SWFMovie->save](#)
- [SWFMovie::saveToFile](#)

- [SWFMovie->setbackground](#)
- [SWFMovie->setdimension](#)
- [SWFMovie->setframes](#)
- [SWFMovie->setrate](#)
- [SWFMovie::startSound](#)
- [SWFMovie::stopSound](#)
- [SWFMovie->streammp3](#)
- [SWFMovie::writeExports](#)
- [SWFMovie](#)
- [SWFPrebuiltClip](#)
- [SWFShape->addFill](#)
- [SWFShape::drawArc](#)
- [SWFShape::drawCircle](#)
- [SWFShape::drawCubic](#)
- [SWFShape::drawCubicTo](#)
- [SWFShape->drawCurve](#)
- [SWFShape->drawCurveTo](#)
- [SWFShape::drawGlyph](#)
- [SWFShape->drawLine](#)
- [SWFShape->drawLineTo](#)
- [SWFShape->movePen](#)
- [SWFShape->movePenTo](#)
- [SWFShape->setLeftFill](#)
- [SWFShape->setLine](#)
- [SWFShape->setRightFill](#)
- [SWFShape](#)
- [SWFSound](#)
- [SWFSoundInstance::loopCount](#)
- [SWFSoundInstance::loopInPoint](#)
- [SWFSoundInstance::loopOutPoint](#)
- [SWFSoundInstance::noMultiple](#)
- [swfsprite->add](#)
- [SWFSprite::labelFrame](#)
- [SWFSprite->nextframe](#)
- [SWFSprite->remove](#)
- [SWFSprite->setframes](#)
- [SWFSprite::startSound](#)
- [SWFSprite::stopSound](#)
- [SWFSprite](#)
- [SWFText->addString](#)
- [SWFText::addUTF8String](#)
- [SWFText::getAscent](#)
- [SWFText::getDescent](#)
- [SWFText::getLeading](#)
- [SWFText::getUTF8Width](#)
- [SWFText->getWidth](#)
- [SWFText->moveTo](#)
- [SWFText->setColor](#)
- [SWFText->setFont](#)
- [SWFText->setHeight](#)
- [SWFText->setSpacing](#)
- [SWFText](#)
- [SWFTextField::addChars](#)
- [SWFTextField->addstring](#)

- [SWFTextField->align](#)
- [SWFTextField->setbounds](#)
- [SWFTextField->setcolor](#)
- [SWFTextField->setFont](#)
- [SWFTextField->setHeight](#)
- [SWFTextField->setindentation](#)
- [SWFTextField->setLeftMargin](#)
- [SWFTextField->setLineSpacing](#)
- [SWFTextField->setMargins](#)
- [SWFTextField->setname](#)
- [SWFTextField::setPadding](#)
- [SWFTextField->setrightMargin](#)
- [SWFTextField](#)
- [SWFVideoStream::getNumFrames](#)
- [SWFVideoStream::setDimension](#)
- [SWFVideoStream](#)

8.73.9 `ming_setcubicthreshold()` : Fixe le niveau de seuil cubique (??)

void `ming_setcubicthreshold` (int threshold)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.10 `ming_setscale()` : Configure l'échelle (??)

void `ming_setscale` (int scale)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.11 `ming_useConstants()` : Utilise une constante de réserve (?)

void `ming_useConstants` (int use)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.12 `ming_useswfversion()` : Utilise la version de SWF (??)

void `ming_useswfversion` (int version)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.13 `SWFAction()` : Crée une nouvelle action

SWFAction `swfaction` (string script)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFAction crée une nouvelle action et compile le script script .

La syntaxe du script est basée sur le langage C, mais celui-ci utilise aussi beaucoup de notions propres à SWF : le bytecode SWF est trop simpliste pour faire l'essentiel de ce que l'on veut. Par exemple, il n'est pas possible de faire des fonctions sans descendre profondément dans les entrailles de la machine, car le bytecode de saut est écrit en dur. Pas moyen de pousser une adresse dans la pile, ou de dépiler - Chaque fonction doit savoir exactement où elle retourne.

Alors, que reste-t-il ? Le compilateur reconnaît les mots suivants :

- break
- for
- continue
- if
- else
- do
- while

Il n'y a pas de type de données : toutes les valeurs de SWF sont stockées comme des chaînes de caractères. Les fonctions suivantes peuvent être utilisées dans les expressions :

time()

Retourne le nombre de milli-secondes depuis le début de l'animation.

random(seed)

Retourne un nombre pseudo-aléatoire, entre 0 et seed.

length(expr)

Retourne la taille de l'expression donnée.

int(number)

Retourne le nombre number, arrondi à l'entier inférieur le plus proche.

concat(expr, expr)

Retourne la concaténation des deux expressions.

ord(expr)

Retourne le code ASCII du caractère expr.

chr(num)

Retourne le caractère pour le code ASCII num.

substr(string, location, length)

Retourne la sous-chaîne, extraite de string , de longueur length et commençant au caractère location .

De plus, les commandes suivantes sont accessibles :

duplicateClip(clip, name, depth)

Duplique le sprite nommé clip . La nouvelle animation a le nouveau nom name et la profondeur depth .

removeClip(expr)

Supprime l'animation nommée expr.

trace(expr)

Ecrit l'expression expr dans le fichier d'historique. Il est peut probable que le navigateur ou le lecteur fasse quoi que ce soit avec.

startDrag(target, lock, [left, top, right, bottom])

Commence à déplacer l'animation target . L'argument lock indique si le déplacement verrouille la souris (utilisez 0, FALSE) ou 1 (TRUE)). Les paramètres optionnels délimitent la zone de déplacement.

stopDrag()

Cesse le déplacement de l'animation.

callFrame(expr)

Appelle l'image expr comme une fonction.

getURL(url, target, [method])

Charge l'URL url dans l'objet target. target correspond à l'élément target du document HTML (comme "_top" ou "_blank"). L'argument optionnel method peut être POST ou GET, si vous voulez envoyer des variables au serveur.

loadMovie(url, target)

Charge l'URL url dans l'objet target. target peut être l'image courante (je pense) ou une des valeurs magiques "_level0" (remplace l'animation courante) ou "_level1" (charge une nouvelle animation à la place de la courante).

nextFrame()

Va à l'image suivante.

prevFrame()

Va à l'image précédente.

play()

Joue l'animation.

stop()

Cesse de jouer l'animation.

toggleQuality()

Passe de haute en basse qualité (et vice-versa).

stopSounds()

Cesse de jouer les sons.

gotoFrame(num)

Va à l'image numéro num . Les images sont numérotées à partir de 0.

gotoFrame(name)

Va à l'image nommée name . Ce qui est carrément cool, car les labels ne sont pas encore supportés pour les images.

setTarget(expr)

Modifie le contexte de l'action. C'est ce qu'ils disent, mais je n'ai pas trop d'idées là-dessus.

Et il y a un truc bizarre : l'expression frameLoaded(num) peut être utilisée dans les conditions if et dans les boucles while pour vérifier si une image a été chargée. En tous cas, c'est ce qu'elle est supposé faire, mais je ne l'ai jamais testée, et je doute sérieusement que cela fonctionne. Vous pouvez utiliser plutôt `framesLoaded` à la place.

Les sprites ont des propriétés. Vous pouvez les lire toutes (vraiment?), en modifier quelques-unes. Les voici :

- x
- y
- xScale
- yScale
- currentFrame - (lecture seule)
- totalFrames - (lecture seule)
- alpha - Niveau de transparence
- visible - 1=on, 0=off (?)
- width - (lecture seule)
- height - (lecture seule)

- rotation
- target - (lecture seule) (???)
- framesLoaded - (lecture seule)
- name
- dropTarget - (lecture seule) (???)
- url - (lecture seule) (???)
- highQuality - 1=high, 0=low (?)
- focusRect - (???)
- soundBufTime - (???)

Modifier la position d'un sprite est aussi simple que `/box.x = 100;` . Pourquoi le slash initial ? C'est comme cela que Flash garde la trace des sprites dans les animations, un peu comme des fichiers sous Unix. Si le sprite qui s'appelle box a lui-même un autre sprite appelé biff, vous pouvez y accéder avec la commande `/box/biff.x = 100;` . En tous cas, ça marche pour moi. Corrigez-moi si c'est faux!

Cet exemple simple va déplacer le gros carré rouge dans la fenêtre.

Exemple avec `swfaction`

```
<?php
  $s = new SWFShape();
  $f = $s->addFill(0xff, 0, 0);
  $s->setRightFill($f);

  $s->movePenTo(-500, -500);
  $s->drawLineTo(500, -500);
  $s->drawLineTo(500, 500);
  $s->drawLineTo(-500, 500);
  $s->drawLineTo(-500, -500);

  $p = new SWFSprite();
  $i = $p->add($s);
  $i->setDepth(1);
  $p->nextFrame();

  for ($n=0; $n<5; ++$n) {
    $i->rotate(-15);
    $p->nextFrame();
  }

  $m = new SWFMovie();
  $m->setBackground(0xff, 0xff, 0xff);
  $m->setDimension(6000, 4000);

  $i = $m->add($p);
  $i->setDepth(1);
  $i->moveTo(-500, 2000);
  $i->setName("box");

  $m->add(new SWFAction("/box.x += 3;"));
  $m->nextFrame();
  $m->add(new SWFAction("gotoFrame(0); play();"));
  $m->nextFrame();

  header('Content-type: application/x-shockwave-flash');
  $m->output();
?>
```

Cet exemple suit votre souris sur l'écran.

Exemple avec `swfaction`

```

<?php

$m = new SWFMovie();
$m->setRate(36.0);
$m->setDimension(1200, 800);
$m->setBackground(0, 0, 0);

/* sprite de suivi de souris : vide, mais il suit la souris
   de manière à ce que nous connaissions ses coordonnées */

$i = $m->add(new SWFSprite());
$i->setName('mouse');

$m->add(new SWFAction("
    startDrag('/mouse', 1); /* '1' signifie verrouiller la souris */
"));

/* On peut tout simplement virer l'anti-aliasing, car il n'y a
   que des gros carrés, finalement */

$m->add(new SWFAction("
    this.quality = 0;
"));

/* boîte de morphing */
$r = new SWFMorph();
$s = $r->getShape1();

/* Notez que ce n'est pas pratique pour les formes habituelles.
   Aucune idée de pourquoi */
$s->setLeftFill($s->addFill(0xff, 0xff, 0xff));
$s->movePenTo(-40, -40);
$s->drawLine(80, 0);
$s->drawLine(0, 80);
$s->drawLine(-80, 0);
$s->drawLine(0, -80);

$s = $r->getShape2();

$s->setLeftFill($s->addFill(0x00, 0x00, 0x00));
$s->movePenTo(-1, -1);
$s->drawLine(2, 0);
$s->drawLine(0, 2);
$s->drawLine(-2, 0);
$s->drawLine(0, -2);

/* sprite contenant la boîte de morphing -
   c'est juste un scénario avec une boîte de morphing */

$box = new SWFSprite();
$box->add(new SWFAction("
    stop();
"));
$i = $box->add($r);

for ($n=0; $n<=20; ++$n) {
    $i->setRatio($n/20);
    $box->nextFrame();
}

/* ce conteneur nous permet d'utiliser la même action plusieurs fois */

$cell = new SWFSprite();
$i = $cell->add($box);
$i->setName('box');

$cell->add(new SWFAction("

```

```

    setTarget('box');

    /* ...x représente les x coordonnées du parent, i.e. (...)x */
    dx = (/mouse.x + random(6)-3 - ...x)/5;
    dy = (/mouse.y + random(6)-3 - ...y)/5;
    gotoFrame(int(dx*dx + dy*dy));

    "));

$cell->nextFrame();
$cell->add(new SWFAction("

    gotoFrame(0);
    play();

    "));

$cell->nextFrame();

/* finalement, ajoutons quelques cellules à l'animation */
for ($x=0; $x<12; ++$x) {
    for ($y=0; $y<8; ++$y) {
        $i = $m->add($cell);
        $i->moveTo(100*$x+50, 100*$y+50);
    }
}

$m->nextFrame();

$m->add(new SWFAction("

    gotoFrame(1);
    play();

    "));

header('Content-type: application/x-shockwave-flash');
$m->output();
?>

```

La même chose que ci-dessus, mais en couleurs.

Exemple avec swfaction

```

<?php

$m = new SWFMovie();
$m->setDimension(11000, 8000);
$m->setBackground(0x00, 0x00, 0x00);

$m->add(new SWFAction("

this.quality = 0;
/frames.visible = 0;
startDrag('/mouse', 1);

    "));

// sprite de suivi de souris
$t = new SWFSprite();
$i = $m->add($t);
$i->setName('mouse');

$g = new SWFGradient();

```

```

$g->addEntry(0, 0xff, 0xff, 0xff, 0xff);
$g->addEntry(0.1, 0xff, 0xff, 0xff, 0xff);
$g->addEntry(0.5, 0xff, 0xff, 0xff, 0x5f);
$g->addEntry(1.0, 0xff, 0xff, 0xff, 0);

// gradient
$s = new SWFShape();
$f = $s->addFill($g, SWFFILL_RADIAL_GRADIENT);
$f->scaleTo(0.03);
$s->setRightFill($f);
$s->movePenTo(-600, -600);
$s->drawLine(1200, 0);
$s->drawLine(0, 1200);
$s->drawLine(-1200, 0);
$s->drawLine(0, -1200);

// on en fait un sprite pour utiliser la fonction multColor()
$p = new SWFSprite();
$p->add($s);
$p->nextFrame();

// Ajoute la forme ici, chaque forme dans une couleur différente
$q = new SWFSprite();
$q->add(new SWFAction("gotoFrame(random(7)+1); stop();"));
$i = $q->add($p);

$i->multColor(1.0, 1.0, 1.0);
$q->nextFrame();
$i->multColor(1.0, 0.5, 0.5);
$q->nextFrame();
$i->multColor(1.0, 0.75, 0.5);
$q->nextFrame();
$i->multColor(1.0, 1.0, 0.5);
$q->nextFrame();
$i->multColor(0.5, 1.0, 0.5);
$q->nextFrame();
$i->multColor(0.5, 0.5, 1.0);
$q->nextFrame();
$i->multColor(1.0, 0.5, 1.0);
$q->nextFrame();

// Enfin, le code de l'action
$p = new SWFSprite();
$i = $p->add($q);
$i->setName('frames');
$p->add(new SWFAction("

dx = (/:mousex-/:lastx)/3 + random(10)-5;
dy = (/:mousey-/:lasty)/3;
x = /:mousex;
y = /:mousey;
alpha = 100;

    "));
$p->nextFrame();

$p->add(new SWFAction("

this.x = x;
this.y = y;
this.alpha = alpha;
x += dx;
y += dy;
dy += 3;
alpha -= 8;

    "));

```

```

$p->nextFrame();

$p->add(new SWFAction("prevFrame(); play();"));
$p->nextFrame();

$i = $m->add($p);
$i->setName('frames');
$m->nextFrame();

$m->add(new SWFAction("

lastx = mousex;
lasty = mousey;
mousex = /mouse.x;
mousey = /mouse.y;

++num;

if (num == 11)
    num = 1;

removeClip('char' & num);
duplicateClip(/frames, 'char' & num, num);

"));

$m->nextFrame();
$m->add(new SWFAction("prevFrame(); play();"));

header('Content-type: application/x-shockwave-flash');
$m->output();
?>

```

8.73.14 SWFBitmap->getHeight() : Retourne la hauteur d'une bitmap Flash

float **swfbitmap->getheight** (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFBitmap->getHeight retourne la hauteur d'une bitmap, en pixels.

Voir aussi SWFBitmap->getWidth .

8.73.15 SWFBitmap->getWidth() : Retourne la largeur d'une bitmap Flash

float **swfbitmap->getwidth** (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFBitmap->getWidth retourne la largeur d'une bitmap, en pixels.

Voir aussi SWFBitmap->getHeight .

8.73.16 SWFBitmap() : Crée un objet bitmap

SWFBitmap **swfbitmap** (mixed file , mixed alphafile)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFBitmap crée un objet bitmap à partir d'un fichier JPEG ou DBL dans le fichier file . alphafile indique un fichier de masque à utiliser comme canal alpha sur une image JPEG. Les deux paramètres peuvent être une ressource ouverte avec fopen ou une chaîne binaire.

Note

Seuls les JPEG baseline (frame 0) sont supportés. Les baseline optimisées ou les JPEG progressives ne sont pas supportées.

SWFBitmap a les méthodes suivantes : SWFBitmap->getWidth et SWFBitmap->getHeight .

Il n'est pas possible d'importer directement des images PNG, il faut utiliser l'utilitaire de conversion png2dbl pour en faire un fichier .dbl ("define bits lossless"). La raison est que l'auteur ne souhaite pas de dépendance avec la bibliothèque PNG. Le fichier d'autoconfiguration devrait régler ce problème, mais il n'est pas encore fait.

Importation de fichiers PNG sous Ming

```

<?php
$s = new SWFShape();
$f = $s->addFill(new SWFBitmap(file_get_contents("png.dbl")));
$s->setRightFill($f);

$s->drawLine(32, 0);
$s->drawLine(0, 32);
$s->drawLine(-32, 0);
$s->drawLine(0, -32);

$m = new SWFMovie();
$m->setDimension(32, 32);
$m->add($s);

header('Content-type: application/x-shockwave-flash');
$m->output();
?>

```

Et vous pouvez ajouter un masque alpha sur une image JPEG.

Exemple avec SWFBitmap

```

<?php

$s = new SWFShape();

//les fichiers .msk sont générés par l'utilitaire "gif2mask"
$f = $s->addFill(new SWFBitmap(file_get_contents("alphafill.jpg"), file_get_contents("alphafill.msk")));
$s->setRightFill($f);

$s->drawLine(640, 0);
$s->drawLine(0, 480);
$s->drawLine(-640, 0);
$s->drawLine(0, -480);

$c = new SWFShape();
$c->setRightFill($c->addFill(0x99, 0x99, 0x99));

```

```

$c->drawLine(40, 0);
$c->drawLine(0, 40);
$c->drawLine(-40, 0);
$c->drawLine(0, -40);

$m = new SWFMovie();
$m->setDimension(640, 480);
$m->setBackground(0xcc, 0xcc, 0xcc);

// décide un fond à damier
for ($y=0; $y<480; $y+=40) {
    for ($x=0; $x<640; $x+=80) {
        $i = $m->add($c);
        $i->moveTo($x, $y);
    }

    $y+=40;

    for ($x=40; $x<640; $x+=80) {
        $i = $m->add($c);
        $i->moveTo($x, $y);
    }
}

$m->add($s);

header('Content-type: application/x-shockwave-flash');
$m->output();
?>

```

8.73.17 SWFbutton->addAction() : Ajoute une action au bouton Flash

void **swfbutton->addaction** (resource action , int flags)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFbutton->addAction ajoute l'action action (créée par SWFAction) au bouton courant, dans les conditions précisées par flags . Les valeurs valides de flags sont : SWFBUTTON_MOUSEOVER , SWFBUTTON_MOUSEOUT , SWFBUTTON_MOUSEUP , SWFBUTTON_MOUSEUPOUTSIDE , SWFBUTTON_MOUSEDOWN , SWFBUTTON_DRAGOUT et SWFBUTTON_DRAGOVER .

Voir aussi SWFbutton->addshape et SWFAction .

8.73.18 SWFButton::addASound() : Associe un son avec une transition de bouton

SWFSoundInstance **SWFButton::addASound** (SWFSound sound , int flags)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.19 SWFbutton->addShape() : Ajoute une forme à un bouton

void **swfbutton->addshape** (resource shape , int flags)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFbutton->addShape ajoute la forme shape au bouton. Les valeurs possibles de flags sont : SWFBUTTON_UP , SWFBUTTON_OVER , SWFBUTTON_DOWN ou SWFBUTTON_HIT . SWFBUTTON_HIT n'est même pas affiché, elle désigne la région du clic d'un bouton. C'est-à-dire que tout point où le bouton est dessiné est considéré comme accessible.

8.73.20 SWFbutton->setAction() : Assigne l'action du bouton

void **swfbutton->setaction** (resource action)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFbutton->setAction assigne l'action qui sera exécutée lorsque le bouton sera cliqué. C'est un alias de la fonction addAction(shape, SWFBUTTON_MOUSEUP) . action est une SWFAction .

Voir aussi SWFbutton->addShape et SWFAction .

8.73.21 SWFbutton->setdown() : Alias de addShape(shape, SWFBUTTON_DOWN)

void **swfbutton->setdown** (resource shape)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFbutton->setdown est un alias pour addShape(shape, SWFBUTTON_DOWN) .

Voir aussi SWFbutton->addShape et SWFAction .

8.73.22 SWFbutton->setHit() : Alias de addShape(shape, SWFBUTTON_HIT)

void **swfbutton->sethit** (resource shape)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFbutton->setHit est un alias pour `addShape(shape, SWFBUTTON_HIT)` .

Voir aussi SWFbutton->addShape et SWFAction .

8.73.23 SWFButton::setMenu() : Active le traçage comme comportement de menu de bouton

void **SWFButton::setMenu** (int flag)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.24 SWFbutton->setOver() : Alias de addShape(shape, SWFBUTTON_OVER)

void **swfbutton->setover** (resource shape)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, **SANS PREAVIS!** Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFbutton->setOver est un alias de la fonction `addShape(shape, SWFBUTTON_OVER)` .

Voir aussi SWFbutton->addShape et SWFAction .

8.73.25 SWFbutton->setUp() : Alias de SWFbutton->addShape(shape, SWFBUTTON_UP)

void **swfbutton->setup** (resource shape)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, **SANS PREAVIS!** Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFbutton->setUp est un alias pour `SWFbutton->addShape(shape, SWFBUTTON_UP)` .

Voir aussi SWFbutton->addShape et SWFAction .

8.73.26 SWFbutton() : Crée un nouveau bouton

SWFButton **swfbutton** (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, **SANS PREAVIS!** Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFbutton crée un nouveau bouton. Cliquez-le, passez la souris dessus, et appelez des actions. Facile !

SWFButton a les méthodes suivantes : [SWFbutton->addShape](#) , [SWFbutton->setUp](#) , [swfbutton->setover](#) [SWFbutton->setDown](#) , [swfbutton->sethit](#) [SWFbutton->setAction](#) et [SWFbutton->addAction](#) .

Cet exemple simplissime vous montre comme faire un roll-over, un roll-on, un clic, un relâché de souris, et rien du tout (pas d'action).

Exemple avec [SWFbutton](#)

```
<?php
$f = new SWFFont("_serif");
$p = new SWFSprite();
function label($string)
{
    global $f;
    $t = new SWFTextField();
    $t->setFont($f);
    $t->addString($string);
    $t->setHeight(200);
    $t->setBounds(3200,200);
    return $t;
}
function addLabel($string)
{
    global $p;
    $i = $p->add(label($string));
    $p->nextFrame();
    $p->remove($i);
}
$p->add(new SWFAction("stop();"));
addLabel("NO ACTION");
addLabel("SWFBUTTON_MOUSEUP");
addLabel("SWFBUTTON_MOUSEDOWN");
addLabel("SWFBUTTON_MOUSEOVER");
addLabel("SWFBUTTON_MOUSEOUT");
addLabel("SWFBUTTON_MOUSEUPOUTSIDE");
addLabel("SWFBUTTON_DRAGOVER");
addLabel("SWFBUTTON_DRAGOUT");
function rect($r, $g, $b)
{
    $s = new SWFShape();
    $s->setRightFill($s->addFill($r, $g, $b));
    $s->drawLine(600,0);
    $s->drawLine(0,600);
    $s->drawLine(-600,0);
    $s->drawLine(0,-600);
    return $s;
}
$b = new SWFButton();
$b->addShape(rect(0xff, 0, 0), SWFBUTTON_UP | SWFBUTTON_HIT);
$b->addShape(rect(0, 0xff, 0), SWFBUTTON_OVER);
$b->addShape(rect(0, 0, 0xff), SWFBUTTON_DOWN);
$b->addAction(new SWFAction("setTarget('/label'); gotoFrame(1);"),
    SWFBUTTON_MOUSEUP);
$b->addAction(new SWFAction("setTarget('/label'); gotoFrame(2);"),
    SWFBUTTON_MOUSEDOWN);
$b->addAction(new SWFAction("setTarget('/label'); gotoFrame(3);"),
    SWFBUTTON_MOUSEOVER);
$b->addAction(new SWFAction("setTarget('/label'); gotoFrame(4);"),
    SWFBUTTON_MOUSEOUT);
$b->addAction(new SWFAction("setTarget('/label'); gotoFrame(5);"),
    SWFBUTTON_MOUSEUPOUTSIDE);
$b->addAction(new SWFAction("setTarget('/label'); gotoFrame(6);"),
    SWFBUTTON_DRAGOVER);
$b->addAction(new SWFAction("setTarget('/label'); gotoFrame(7);"),
    SWFBUTTON_DRAGOUT);
```

```

$m = new SWFMovie();
$m->setDimension(4000,3000);
$i = $m->add($p);
$i->setName("label");
$i->moveTo(400,1900);
$i = $m->add($b);
$i->moveTo(400,900);
header('Content-type: application/x-shockwave-flash');
$m->output();
?>

```

Cet exemple simple illustre le déplacement d'un gros bouton rouge dans la fenêtre. Ce n'est pas du tirer-déposer, mais juste du tirer.

Exemple avec `SWFbutton->addAction`

```

<?php
$s = new SWFShape();
$s->setRightFill($s->addFill(0xff, 0, 0));
$s->drawLine(1000,0);
$s->drawLine(0,1000);
$s->drawLine(-1000,0);
$s->drawLine(0,-1000);
$b = new SWFButton();
$b->addShape($s, SWFBUTTON_HIT | SWFBUTTON_UP | SWFBUTTON_DOWN | SWFBUTTON_OVER);
$b->addAction(new SWFAction("startDrag('/test', 0);"),
    SWFBUTTON_MOUSEDOWN);
// '0' signifie : ne pas verrouiller la souris
$b->addAction(new SWFAction("stopDrag();"),
    SWFBUTTON_MOUSEUP | SWFBUTTON_MOUSEUPOUTSIDE);
$p = new SWFSprite();
$p->add($b);
$p->nextFrame();
$m = new SWFMovie();
$i = $m->add($p);
$i->setName('test');
$i->moveTo(1000,1000);
header('Content-type: application/x-shockwave-flash');
$m->output();
?>

```

8.73.27 SWFDisplayItem::addAction() : Ajoute cet SWFAction à l'instance donnée SWFSprite

void **SWFDisplayItem::addAction** (SWFAction action , int flags)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.28 SWFDisplayItem->addColor() : Ajoute une couleur à une transformation

void **swfdisplayitem->addcolor** (int red , int green , int blue , int a)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, **SANS PREAVIS!** Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFDisplayItem->addColor ajoute une couleur à la transformations courante. La couleur est donnée sous la forme RGB.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie->add .

8.73.29 SWFDisplayItem::endMask() : Un autre moyen pour définir une couche MASK

void **SWFDisplayItem::endMask** (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.30 SWFDisplayItem::getRot() :

float **SWFDisplayItem::getRot** (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.31 SWFDisplayItem::getX() :

float **SWFDisplayItem::getX** (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.32 SWFDisplayItem::getXScale() :

float **SWFDisplayItem::getXScale** (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.33 SWFDisplayItem::getXSkew() :

float **SWFDisplayItem::getXSkew** (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.34 SWFDisplayItem::getY() :

float **SWFDisplayItem::getY** (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.35 SWFDisplayItem::getYScale() :float **SWFDisplayItem::getYScale** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.36 SWFDisplayItem::getYSkew() :float **SWFDisplayItem::getYSkew** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.37 SWFDisplayItem->move() : Déplace un objet en coordonnées relativesvoid **swfdisplayitem->move** (int dx , int dy)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFDisplayItem->move déplace la forme courante de dx et dy unités, depuis sa position courante.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie->add .

Voir aussi SWFDisplayItem->moveTo .**8.73.38 SWFDisplayItem->moveTo() : Déplace un objet en coordonnées globales**void **swfdisplayitem->moveto** (int x , int y)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFDisplayItem->moveTo déplace la forme courante jusqu'au point de coordonnées globales (x , y).

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie->add .

Voir aussi SWFDisplayItem->move .

8.73.39 SWFDisplayItem->multColor() : Multiplie la couleur de transformation

void **swfdisplayitem->multcolor** (int red , int green , int blue , int a)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFDisplayItem->multColor multiplie la couleur de transformation par les valeurs données.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie->add .

Cet exemple simple modifie l'atmosphère de votre image, et en fait une scène d'Halloween (utilisez un paysage ou une image claire pour un meilleur effet)

Exemple avec SWFDisplayItem->multColor

```
<?php
$b = new SWFBitmap(file_get_contents("backyard.jpg"));
// Utilisez une de vos images
$s = new SWFShape();
$s->setRightFill($s->addFill($b));
$s->drawLine($b->getWidth(), 0);
$s->drawLine(0, $b->getHeight());
$s->drawLine(-$b->getWidth(), 0);
$s->drawLine(0, -$b->getHeight());
$m = new SWFMovie();
$m->setDimension($b->getWidth(), $b->getHeight());
$i = $m->add($s);
for($n=0; $n<=20; ++$n)
{
    $i->multColor(1.0-$n/10, 1.0, 1.0);
    $i->addColor(0xff*$n/20, 0, 0);
    $m->nextFrame();
}
header('Content-type: application/x-shockwave-flash');
$m->output();
?>
```

8.73.40 SWFDisplayItem->remove() : Supprime un objet d'une animation

void **swfdisplayitem->remove** (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFDisplayItem->remove supprime cet objet de la liste d'affichage.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie->add .

Voir aussi SWFMovie->add .

8.73.41 SWFDisplayItem->Rotate() : Fait tourner une forme relativement

void **swfdisplayitem->rotate** (float ddegrees)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFDisplayItem->Rotate fait tourner la forme de ddegrees degrés, en plus de sa rotation courante.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie->add .

Voir aussi SWFDisplayItem->rotateTo .

8.73.42 SWFDisplayItem->rotateTo() : Tourne un objet en angle absolu

void **swfdisplayitem->rotateto** (float degrees)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFDisplayItem->rotateTo tourne l'objet jusqu'à l'angle absolu degrees , en degrés.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie->add .

Cet exemple amène trois chaînes tournoyantes depuis le fond de l'écran. Plutôt sympa.

Exemple avec SWFDisplayItem->rotateTo

```

<?php
$thetext = "ming!";
$f = new SWFFont("Bauhaus 93.fdb");
$m = new SWFMovie();
$m->setRate(24.0);
$m->setDimension(2400, 1600);
$m->setBackground(0xff, 0xff, 0xff);
// Les fonctions avec un nombre d'arguments sont vraiment une bonne idée.
// Sincèrement!
function text($r, $g, $b, $a, $rot, $x, $y, $scale, $string)
{
    global $f, $m;
    $t = new SWFText();
    $t->setFont($f);
    $t->setColor($r, $g, $b, $a);
    $t->setHeight(960);
    $t->moveTo(-($f->getWidth($string))/2, $f->getAscent()/2);
    $t->addString($string);
    // On peut ajouter des propriétés comme pour une variable PHP standard
    // tant que les noms ne sont pas déjà pris.
    // e.g., vous ne pouvez pas utiliser $i->scale, car c'est une fonction.
    $i = $m->add($t);
    $i->x = $x;
    $i->y = $y;
    $i->rot = $rot;
    $i->s = $scale;
}

```

```

    $i->rotateTo($rot);
    $i->scale($scale, $scale);
    // mais les modifications sont locales à une fonction, donc il faut
    // retourner l'objet modifié. Pas pratique...
    return $i;
}
function step($i)
{
    $oldrot = $i->rot;
    $i->rot = 19*$i->rot/20;
    $i->x = (19*$i->x + 1200)/20;
    $i->y = (19*$i->y + 800)/20;
    $i->s = (19*$i->s + 1.0)/20;
    $i->rotateTo($i->rot);
    $i->scaleTo($i->s, $i->s);
    $i->moveTo($i->x, $i->y);
    return $i;
}
// Alors? &Ccedil;a valait la peine, non?
$i1 = text(0xff, 0x33, 0x33, 0xff, 900, 1200, 800, 0.03, $thetext);
$i2 = text(0x00, 0x33, 0xff, 0x7f, -560, 1200, 800, 0.04, $thetext);
$i3 = text(0xff, 0xff, 0xff, 0x9f, 180, 1200, 800, 0.001, $thetext);
for($i=1; $i<=100; ++$i)
{
    $i1 = step($i1);
    $i2 = step($i2);
    $i3 = step($i3);
    $m->nextFrame();
}
header('Content-type: application/x-shockwave-flash');
$m->output();
?>

```

Voir aussi [swfdisplayitem->rotate](#) .

8.73.43 SWFDisplayItem->scale() : Etire un objet relativement

void **swfdisplayitem->scale** (int dx , int dy)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[SWFDisplayItem->scale](#) étire un objet de (dx , dy), à partir de sa taille courante.

L'objet peut être [swfshape](#) , [swfbutton](#) , [swftext](#) ou [swfsprite](#) . Il doit avoir été ajouté à une animation avec la fonction [swfmovie->add](#) .

Voir aussi [SWFDisplayItem->scaleTo](#) .

8.73.44 SWFDisplayItem->scaleTo() : Etire un objet en coordonnées globales

void **swfdisplayitem->scalet**o (int x , int y)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`SWFDisplayItem->scaleTo` étire un objet jusqu'aux dimensions (x , y).

L'objet peut être `swfshape` , `swfbutton` , `swftext` ou `swfsprite` . Il doit avoir été ajouté à une animation avec la fonction `SWFMovie->add` .

Voir aussi `SWFDisplayItem->scale` .

8.73.45 `SWFDisplayItem->setDepth()` : Modifie la place en profondeur (z-order)

void `swfdisplayitem->setdepth` (float depth)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`SWFDisplayItem->setDepth` place l'objet à la profondeur `depth` . Par défaut, l'objet est placé au niveau où il a été ajouté dans l'animation. Les objets les plus anciens sont placés tout en bas, et les nouveaux sont superposés.

L'objet peut être `swfshape` , `swfbutton` , `swftext` ou `swfsprite` . Il doit avoir été ajouté à une animation avec la fonction `swfmovie->add` .

8.73.46 `SWFDisplayItem::setMaskLevel()` : Définit une couche MASK à un niveau

void `SWFDisplayItem::setMaskLevel` (int level)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.47 `SWFDisplayItem::setMatrix()` : Fixe la matrice de transformation de l'item

void `SWFDisplayItem::setMatrix` (float a , float b , float c , float d , float x , float y)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.48 `SWFDisplayItem->setName()` : Nomme un objet

void `swfdisplayitem->setname` (string name)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS

PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFDisplayItem->setName donne à l'objet courant le nom de name . Cela servira à repérer les acteurs d'un script d'action. Cela ne sert qu'avec les sprites.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie->add .

8.73.49 SWFDisplayItem->setRatio() : Modifie le ratio de l'objet

void **swfdisplayitem->setratio** (float ratio)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFDisplayItem->setRatio modifie le ratio de l'objet, et le fixe à ratio . Uniquement utile pour les morphings.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie->add .

Cet exemple simple effectue un morphing délicat de trois cercles concentriques.

Exemple swfdisplayitem->setname

```
<?php
$p = new SWFMorph();
$g = new SWFGradient();
$g->addEntry(0.0, 0, 0, 0);
$g->addEntry(0.16, 0xff, 0xff, 0xff);
$g->addEntry(0.32, 0, 0, 0);
$g->addEntry(0.48, 0xff, 0xff, 0xff);
$g->addEntry(0.64, 0, 0, 0);
$g->addEntry(0.80, 0xff, 0xff, 0xff);
$g->addEntry(1.00, 0, 0, 0);
$s = $p->getShape1();
$f = $s->addFill($g, SWFFILL_RADIAL_GRADIENT);
$f->scaleTo(0.05);
$s->setLeftFill($f);
$s->movePenTo(-160, -120);
$s->drawLine(320, 0);
$s->drawLine(0, 240);
$s->drawLine(-320, 0);
$s->drawLine(0, -240);
$g = new SWFGradient();
$g->addEntry(0.0, 0, 0, 0);
$g->addEntry(0.16, 0xff, 0, 0);
$g->addEntry(0.32, 0, 0, 0);
$g->addEntry(0.48, 0, 0xff, 0);
$g->addEntry(0.64, 0, 0, 0);
$g->addEntry(0.80, 0, 0, 0xff);
$g->addEntry(1.00, 0, 0, 0);
$s = $p->getShape2();
$f = $s->addFill($g, SWFFILL_RADIAL_GRADIENT);
$f->scaleTo(0.05);
$f->skewXTo(1.0);
$s->setLeftFill($f);
$s->movePenTo(-160, -120);
$s->drawLine(320, 0);
$s->drawLine(0, 240);
```

```

$s->drawLine(-320, 0);
$s->drawLine(0, -240);
$m = new SWFMovie();
$m->setDimension(320, 240);
$i = $m->add($p);
$i->moveTo(160, 120);
for($n=0; $n<=1.001; $n+=0.01)
{
    $i->setRatio($n);
    $m->nextFrame();
}
header('Content-type: application/x-shockwave-flash');
$m->output();
?>

```

8.73.50 SWFDisplayItem->skewX() : Incline suivant les X relativement

void **swfdisplayitem->skewx** (float ddegrees)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFDisplayItem->skewX ajoute ddegrees à l'inclinaison courante (x-skew).

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie->add .

Voir aussi SWFDisplayItem->skewx , SWFDisplayItem->skewy et SWFDisplayItem->skewyto .

8.73.51 SWFDisplayItem->skewXTo() : Incline suivant les X

void **swfdisplayitem->skewxto** (float degrees)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFDisplayItem->skewXTo modifie l'inclinaison (x-skew) à degrees . Si degrees vaut 1.0, l'angle sera de 45°, en avant. S'il vaut plus, ce sera plus penché, et s'il vaut moins, ce sera plus droit.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie->add .

Voir aussi SWFDisplayItem->skewx , SWFDisplayItem->skewy et SWFDisplayItem->skewyto .

8.73.52 SWFDisplayItem->skewY() : Incline suivant les Y relativement

void **swfdisplayitem->skewy** (float ddegrees)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFDisplayItem->skewY ajoute degrees à l'inclinaison courante (y-skew).

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie->add .

Voir aussi SWFDisplayItem->skewyto , SWFDisplayItem->skewx et SWFDisplayItem->skewxto .

8.73.53 SWFDisplayItem->skewYTo() : Incline suivant les Y

void **swfdisplayitem->skewyto** (float degrees)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFDisplayItem->skewYTo modifie l'inclinaison (y-skew) à degrees . Si degrees vaut 1.0, l'angle sera de 45°, en haut. S'il vaut plus, ce sera plus penché, et s'il vaut moins, ce sera plus droit.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie->add .

Voir aussi SWFDisplayItem->skewy , SWFDisplayItem->skewx et SWFDisplayItem->skewxto .

8.73.54 SWFFill->moveTo() : Déplace l'origine de l'objet SWFFill

void **swffill->moveto** (int x , int y)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFFill->moveTo déplace l'origine de la forme jusqu'au point de coordonnées globales (x , y) .

8.73.55 SWFFill->rotateTo() : Tourne la forme

void **swffill->rotateto** (float degrees)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFFill->rotateTo tourne la forme depuis son orientation initiale jusqu'à un angle de degrees degrés.

8.73.56 SWFFill->scaleTo() : Modifie l'échelle de la forme

void **swffill->scalet**o (int x , int y)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFFill->scaleTo modifie l'échelle de la forme de x dans le sens des abscisses et y dans le sens des ordonnées.

8.73.57 SWFFill->skewXTo() : Incline (abscisses)

void **swffill->skewx**to (float x)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFFill->skewXTo incline la forme de x suivant l'axe des abscisses. Si x vaut 1.0, l'inclinaison sera de 45° degrés, en avant. Si x vaut plus, l'inclinaison sera plus forte et, sinon, la forme sera plus droite.

8.73.58 SWFFill->skewYTo() : Incline (ordonnées)

void **swffill->skewy**to (float y)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFFill->skewYTo incline la forme de y suivant l'axe des abscisses. Si y vaut 1.0, l'inclinaison sera de 45° degrés, en avant. Si x vaut plus, l'inclinaison sera plus forte, et sinon, la forme sera plus droite.

8.73.59 SWFFill() : Crée un objet de remplissage

SWFFill **swffill** (void)

SWFFill vous permet de transformer une image bitmap ou un gradient. Les objets swffill sont créés par swfshape->addfill .

SWFFill a les méthodes suivantes : swffill->moveto , swffill->scaleto , swffill->rotateto , swffill->skewxto et swffill->skewyto .

8.73.60 SWFFont::getAscent() : Retourne la hampe de la police ou 0 si elle est non disponible

float **SWFFont::getAscent** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.61 SWFFont::getDescent() : Retourne le jambage de la police ou 0 s'il n'est pas disponible

float **SWFFont::getDescent** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.62 SWFFont::getLeading() : Retourne l'interlignage de la police ou 0 s'il n'est pas disponible

float **SWFFont::getLeading** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.63 SWFFont::getShape() : Retourne la forme glyphe d'un caractère en tant que chaîne de caractères

string **SWFFont::getShape** (int code)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.64 SWFFont::getUTF8Width() : Calcule la largeur de la chaîne de caractères donnée dans cette police à sa pleine taille

float **SWFFont::getUTF8Width** (string string)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.65 swffont->getwidth() : Retourne la taille de la chaîne

float **swffont->getwidth** (string string)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[swffont->getwidth](#) retourne la taille de la chaîne string , avec la police courante. Vous utiliserez plutôt la même méthode de l'objet [swftext](#) , qui utilise les paramètres de l'objet.

8.73.66 SWFFont() : Charge une police

SWFFont **swffont** (string filename)

Attention
 Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

si filename est le nom d'un fichier FDB (i.e., si le nom de fichier se termine par ".fdb"), charge la police.

FDB ("font definition block") est un petit utilitaire pour Flash DefineFont2 qui contient une description complète de la police. Vous pouvez créer des fichiers FDB à partir du "SWT Generator", qui est inclus avec les utilitaires makefdb - regardez dans le dossier utilitaire de Ming.

Les polices utilisateurs ne contiennent aucune information autre que le nom de la police. On suppose que la police sera elle-même accessible au lecteur. Les polices "_serif", "_sans", et "_typewriter" doivent être universellement disponibles. Par exemple :

```
<?php
$f = newSWFFont ("_sans");
?>
```

vous donne la police standard "sans-serif", probablement identique à celle que vous obtenez avec le code .

[swffont](#) retourne une ressource de police, à utiliser avec les méthodes [swftext->setfont](#) et [swftextfield->setfont](#) .

SWFFont a les méthodes suivantes : [SWFFont->getwidth](#) .

8.73.67 SWFFontChar::addChars() : Ajoute des caractères à une police pour exporter une police

void **SWFFontChar::addChars** (string char)

Attention
 Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.68 SWFFontChar::addUTF8Chars() : Ajoute des caractères à une police pour exporter une police

void **SWFFontChar::addUTF8Chars** (string char)

Attention
 Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.69 SWFGradient->addEntry() : Ajoute une couleur à la liste du gradient

void **swfgradient->addentry** (float ratio , int red , int green , int blue , *int a*)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFGradient->addEntry ajoute une couleur à la liste des couleurs du gradient. ratio est un nombre de 0 à 1, qui indique l'ordre d'apparition des couleurs. Vous devez ajouter les couleurs dans l'ordre croissant de ratio.

red , green , blue représente une couleur, au format RGB. Le dernier paramètre a est optionnel.

8.73.70 SWFGradient() : Crée un objet gradient

SWFGradient **swfgradient** (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFGradient crée un nouvel objet gradient.

Une fois que vous avez ajouté les couleurs à votre gradient, vous pouvez l'utiliser dans des formes, avec la fonction swfshape->addfill .

SWFGradient a la méthode suivante : swfgradient->addentry .

Cet exemple simple affiche un gradient noir-blanc comme fond, et un gradient concentrique au centre.

Exemple avec SWFGradient

```
<?php
$m = new SWFMovie();
$m->setDimension(320, 240);
$s = new SWFShape();
// gradient noir-blanc
$g = new SWFGradient();
$g->addEntry(0.0, 0, 0, 0);
$g->addEntry(1.0, 0xff, 0xff, 0xff);
$f = $s->addFill($g, SWFFILL_LINEAR_GRADIENT);
$f->scaleTo(0.01);
$f->moveTo(160, 120);
$s->setRightFill($f);
$s->drawLine(320, 0);
$s->drawLine(0, 240);
$s->drawLine(-320, 0);
$s->drawLine(0, -240);
$m->add($s);
$s = new SWFShape();
// gradient radial : rouge vers transparent
$g = new SWFGradient();
$g->addEntry(0.0, 0xff, 0, 0, 0xff);
$g->addEntry(1.0, 0xff, 0, 0, 0);
```

```

$f = $s->addFill($g, SWFFILL_RADIAL_GRADIENT);
$f->scaleTo(0.005);
$f->moveTo(160, 120);
$s->setRightFill($f);
$s->drawLine(320, 0);
$s->drawLine(0, 240);
$s->drawLine(-320, 0);
$s->drawLine(0, -240);
$m->add($s);
header('Content-type: application/x-shockwave-flash');
$m->output();
?>

```

8.73.71 SWFMorph->getshape1() : Sélectionne la forme de départ

mixed **swfmorph->getshape1** (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFMorph->getshape1 sélectionne la forme de début de morphing. swfmorph->getshape1 retourne un objet swfshape .

8.73.72 SWFMorph->getshape2() : Sélectionne la forme de fin

mixed **swfmorph->getshape2** (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFMorph->getshape2 sélectionne la forme de début de morphing. SWFMorph->getshape2 retourne un objet swfshape .

8.73.73 SWFMorph() : Crée un morphing

SWFMorph **swfmorph** (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFMorph crée un morphing.

swfmorph s'appelle aussi "shape tween". C'est cet objet qui permet toutes ces superbes animations qui mettent à genou votre ordinateur. Joie !

Les méthodes ici sont plutôt bizarres. Il serait tellement plus logique d'avoir seulement new SWFMorph(shape1, shape2); , mais, telles que sont les choses aujourd'hui, la deuxième forme a besoin de savoir qu'elle est l'aboutissement d'un morphing. (cel est dû au fait que Flash commence à dessiner aussitôt qu'il a les commandes de dessin. S'il conservait les descriptions de ses propres

formes, et attendait leur totalité avant d'écrire, ceci et bien d'autres choses seraient tellement plus simples).

SWFMorph a les méthodes suivantes : [swfmorph->getshape1](#) et [swfmorph->getshape2](#) .

Cet exemple simple effectue le morphing d'une gros carré rouge en un carré plus petit, bleu et bordé de noir.

Exemple avec [swfmorph](#)

```
<?php
$p = new SWFMorph();
$s = $p->getShape1();
$s->setLine(0,0,0,0);
/* Notez que cela se fait dans l'ordre inverse de l'ordre habituel
   (gauche au lieu de droite), mais je n'ai aucune idée de pourquoi... */
$s->setLeftFill($s->addFill(0xff, 0, 0));
$s->movePenTo(-1000,-1000);
$s->drawLine(2000,0);
$s->drawLine(0,2000);
$s->drawLine(-2000,0);
$s->drawLine(0,-2000);
$s = $p->getShape2();
$s->setLine(60,0,0,0);
$s->setLeftFill($s->addFill(0, 0, 0xff));
$s->movePenTo(0,-1000);
$s->drawLine(1000,1000);
$s->drawLine(-1000,1000);
$s->drawLine(-1000,-1000);
$s->drawLine(1000,-1000);
$m = new SWFMovie();
$m->setDimension(3000,2000);
$m->setBackground(0xff, 0xff, 0xff);
$i = $m->add($p);
$i->moveTo(1500,1000);
for($r=0.0; $r<=1.0; $r+=0.1)
{
    $i->setRatio($r);
    $m->nextFrame();
}
header('Content-type: application/x-shockwave-flash');
$m->output();
?>
```

8.73.74 SWFMovie->add() : Ajoute un objet dans une animation

void **swfmovie->add** (resource instance)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[SWFMovie->add](#) ajoute l'objet instance dans l'animation courante. instance peut être de n'importe quel type : forme (shape), texte (text), police (font), etc. Ils doivent être ajoutés à une animation pour être utilisés.

Pour les objets affichables (formes, textes, boutons, sprites), [swfmovie->add](#) retourne un objet [swfdisplayitem](#) de la liste d'affichage. Ainsi, vous pouvez ajouter la même forme plusieurs fois dans la même animation, et obtenir des ressources différentes pour chaque instance.

Voir aussi tous les autres objets et [swfmovie->remove](#)

Des exemples d'utilisation dans : [swfdisplayitem->rotateto](#) et [swfshape->addfill](#) .

8.73.75 SWFMovie::addExport() :

void **SWFMovie::addExport** (SWFCharacter char , string name)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.76 SWFMovie::addFont() :

SWFFontChar **SWFMovie::addFont** (SWFFont font)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.77 SWFMovie::importChar() :

SWFSprite **SWFMovie::importChar** (string libswf , string name)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.78 SWFMovie::importFont() :

SWFFontChar **SWFMovie::importFont** (string libswf , string name)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.79 SWFMovie::labelFrame() : Écrit un cadre

void **SWFMovie::labelFrame** (string label)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.80 SWFMovie->nextframe() : Passe à l'image suivante

void **swfmovie->nextframe** (void)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
SWFMovie->nextframe passe à l'image suivante de l'animation.

8.73.81 SWFMovie->output() : Envoie votre animation au navigateur

int **swfmovie->output** (*int* *compression*)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFMovie->output envoie votre animation au navigateur. En PHP, faites-le précéder de la fonction header .

```
<?php
header('Content-type: application/x-shockwave-flash');
?>
```

Cela indique au navigateur que l'animation qui arrive est en Flash.

Le paramètre *compression* est un niveau de compression et doit être compris entre 0 et 9. Il définit la compression swf et est similaire à la compression gzip.

Voir aussi SWFMovie->save .

Des exemples d'utilisation dans : SWFMovie->streammp3 , swfdisplayitem->rotateto , swfaction ...
En fait, tous les exemples utilisent cette méthode.

8.73.82 swfmovie->remove() : Supprime un objet d'une animation

void **swfmovie->remove** (*resource* *instance*)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

swfmovie->remove supprime l'objet instance de la liste d'affichage, pour l'animation courante. L'objet ne sera plus disponible pour être affiché ou utilisé.

Voir aussi swfmovie->add .

8.73.83 SWFMovie->save() : Sauve dans un fichier

int **swfmovie->save** (*string* *filename* , *int* *compression*)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFMovie->save sauve votre animation dans le fichier *filename* .

Le paramètre *compression* est un niveau de compression et doit être compris entre 0 et 9. Il définit la compression swf et est similaire à la compression gzip.

Voir aussi [SWFMovie->output](#) .

8.73.84 SWFMovie::saveToFile() :

int **SWFMovie::saveToFile** (stream x , int compression)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.85 SWFMovie->setbackground() : Modifie la couleur de fond

void **swfmovie->setbackground** (int red , int green , int blue)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[SWFMovie->setbackground](#) modifie la couleur de fond. Pourquoi est-ce que cette fonction n'accepte pas de canal alpha ? (réfléchissez quelques instants :-). En fait, cela ne serait pas si stupide : vous pouvez laisser apercevoir le fond HTML à travers l'animation. Il y a un moyen de faire cela, mais cela ne fonctionne qu'avec IE 4. Recherchez sur le site de <http://www.macromedia.com/> pour plus de détails.

8.73.86 SWFMovie->setdimension() : Modifie les dimensions de l'animation

void **swfmovie->setdimension** (int width , int height)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[SWFMovie->setdimension](#) modifie les dimensions de l'animation : width est la largeur et height la hauteur.

8.73.87 SWFMovie->setframes() : Modifie le nombre total d'images dans l'animation

void **swfmovie->setframes** (string numeroofframes)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[SWFMovie->setframes](#) modifie le nombre total d'images dans l'animation, et le fixe à numeroofframes .

8.73.88 SWFMovie->setrate() : Modifie la vitesse de l'animation

```
void swfmovie->setrate ( int rate )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFMovie->setrate fixe la vitesse de l'animation à rate images par secondes. L'animation ralentira d'elle-même si le lecteur Flash ne peut pas afficher suffisamment rapidement, à moins qu'il y ait du son en stream, auquel cas les images sont sacrifiées pour garder un son fluide.

8.73.89 SWFMovie::startSound() :

```
SWFSoundInstance SWFMovie::startSound ( SWFSound sound )
```

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.90 SWFMovie::stopSound() :

```
void SWFMovie::stopSound ( SWFSound sound )
```

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.91 SWFMovie->streammp3() : Envoie un fichier MP3 en streaming

```
void swfmovie->streammp3 ( mixed mp3File )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFMovie->streammp3 envoie le fichier MP3 mp3File en stream audio. swfmovie->streammp3 n'est pas très robuste, et se prend facilement les pieds dans le tapis (elle peut éviter la balise initiale ID3, mais c'est bien tout). Tout comme swfshape->addjpegfill(), ce n'est pas une fonction stable. Il faudra sûrement faire un objet séparé, pour gérer les types de son. Le paramètre mp3File peut être une ressource ouverte avec la fonction fopen ou une chaîne binaire.

Notez que l'animation n'est pas suffisamment intelligente pour ajouter un nombre suffisant d'images, afin de correspondre à la durée totale du stream MP3. Il vous faudra ajouter des images jusqu'à durée de la musique multiplié par le nombre d'images par secondes.

Oui, vous pouvez utiliser Ming pour mettre un rock-'n-roll endiablé dans vos animations. Evitez d'en parler à la RIAA ou la SACEM.

Exemple avec SWFMovie->streammp3

```
<?php
$m = new SWFMovie();
$m->setRate(12.0);
```

```
$m->streamMp3(file_get_contents("distortobass.mp3"));
// utilisez vos propres MP3
// assurez-vous d'avoir les droits
// 11.85 secondes avec 12.0 images par seconde = 142 frames
$m->setFrames(142);
header('Content-type: application/x-shockwave-flash');
$m->output();
?>
```

8.73.92 SWFMovie::writeExports() :

void **SWFMovie::writeExports** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.93 SWFMovie() : Crée un objet 'animation'

SWFMovie **swfmovie** (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFMovie Crée un objet 'animation', représentant une animation Flash version 4.

SWFMovie a les méthodes suivantes : SWFMovie->output , SWFMovie->save , SWFMovie->add , SWFMovie->remove , SWFMovie->nextframe , SWFMovie->setbackground , SWFMovie->setrate , SWFMovie->setdimension , SWFMovie->setframes et SWFMovie->streammp3 .

Des exemples d'utilisation dans : swfdisplayitem->rotateto , swfshape->setline , swfshape->addfill ...
En fait, tous les exemples utilisent cet objet.

8.73.94 SWFPrebuiltClip() : Retourne un objet SWFPrebuiltClip

SWFPrebuiltClip **SWFPrebuiltClip** (*string file*)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.95 SWFShape->addFill() : Ajoute un remplissage plein à la forme Flash

SWFFill **SWFShape->addFill** (int red , int green , int blue , *int a*) SWFFill
SWFShape->addFill (SWFBitmap bitmap , *int flags*) SWFFill **SWFShape->addFill** (SWFGradient gradient , *int flags*)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFShape->addFill ajoute un remplissage plein à la forme. SWFShape->addFill accepte trois différents types d'arguments.

red , green , blue est une couleur (format RGB). Le dernier paramètre a est optionnel.

L'argument bitmap est un objet SWFBitmap . Le paramètre flags peut être l'un des suivants : SWFFILL_CLIPPED_BITMAP ou SWFFILL_TILED_BITMAP. Par défaut, c'est SWFFILL_TILED_BITMAP pour SWFBitmap et SWFFILL_LINEAR_GRADIENT pour SWFGradient.

L'argument gradient est un objet SWFGradient . L'argument flags peut alors prendre l'une des valeurs suivantes : SWFFILL_RADIAL_GRADIENT ou SWFFILL_LINEAR_GRADIENT. Par défaut, c'est SWFFILL_LINEAR_GRADIENT.

SWFShape->addFill retourne un objet swfill à utiliser avec swfshape->setLeftfill , et SWFShape->setRightFill décrite un peu plus loin.

Ceci est un exemple simple qui affiche un cadre sur une bitmap. Ah, il y a un petit bogue dans le lecteur Flash : il ne semble pas faire grand cas de la transformation de la seconde forme en morphing. Suivant les specs, la bitmap devrait s'étirer avec la forme dans cet exemple...

Exemple avec SWFShape->addFill

```
<?php
$p = new SWFMorph();

$b = new SWFBitmap(file_get_contents("alphafill.jpg"));
// utilisez vos propres bitmaps!
$width = $b->getWidth();
$height = $b->getHeight();

$s = $p->getShape1();
$f = $s->addFill($b, SWFFILL_TILED_BITMAP);
$f->moveTo(-$width/2, -$height/4);
$f->scaleTo(1.0, 0.5);
$s->setLeftFill($f);
$s->movePenTo(-$width/2, -$height/4);
$s->drawLine($width, 0);
$s->drawLine(0, $height/2);
$s->drawLine(-$width, 0);
$s->drawLine(0, -$height/2);

$s = $p->getShape2();
$f = $s->addFill($b, SWFFILL_TILED_BITMAP);

// ces déplacements n'ont aucun effet
$f->moveTo(-$width/4, -$height/2);
$f->scaleTo(0.5, 1.0);

$s->setLeftFill($f);
$s->movePenTo(-$width/4, -$height/2);
$s->drawLine($width/2, 0);
$s->drawLine(0, $height);
$s->drawLine(-$width/2, 0);
$s->drawLine(0, -$height);

$m = new SWFMovie();
$m->setDimension($width, $height);
$i = $m->add($p);
$i->moveTo($width/2, $height/2);

for($n=0; $n<1.001; $n+=0.03) {
    $i->setRatio($n);
}
```

```

    $m->nextFrame();
}
header('Content-type: application/x-shockwave-flash');
$m->output();
?>

```

Voir aussi [SWFShape->setLeftFill](#) et [SWFShape->setRightFill](#) .

8.73.96 SWFShape::drawArc() : Dessine un arc de radius r centré à la position courante, d'un angle startAngle à endAngle mesuré dans le sens des aiguilles d'une montre à partir de midi

void **SWFShape::drawArc** (float r , float startAngle , float endAngle)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.97 SWFShape::drawCircle() : Dessine un cercle de radius r centré à la position courante, dans le sens inverse des aiguilles d'une montre

void **SWFShape::drawCircle** (float r)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.98 SWFShape::drawCubic() : Dessine une courbe de Bézier cubique en utilisant la position courante et les trois points donnés en tant que position de contrôle

int **SWFShape::drawCubic** (float bx , float by , float cx , float cy , float dx , float dy)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.99 SWFShape::drawCubicTo() : Dessine une courbe de Bézier cubique en utilisant la position courante et les trois points donnés en tant que points de contrôle

int **SWFShape::drawCubicTo** (float bx , float by , float cx , float cy , float dx , float dy)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.100 SWFShape->drawCurve() : Dessine une courbe relativement

int **swfshape->drawcurve** (int controldx , int controldy , int anchordx , int anchordy , int targetdx , int targetdy)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[SWFShape->drawCurve](#) dessine une courbe quadratique (avec le style courant de ligne, modifié par [swfshape->setline](#)) depuis le point courant jusqu'au point (`anchordx` , `anchordy`) relativement au point courant, et en utilisant le point de contrôle (`controldx` , `controldy`). C'est-à-dire qu'il commence en allant vers le point de contrôle, puis se dirige sur le point d'ancrage.

Avec 6 paramètres, la fonction dessine un cube de bézier du point (`x+ targetdx` , `x+ targetdy`) avec les points de contrôle (`x+ controldx` , `y+ controldy`) et (`x+ anchordx` , `y+ anchordy`).

Voir aussi [SWFShape->drawlineto](#) , [SWFShape->drawline](#) , [SWFShape->movepen](#) et [SWFShape->movepen](#) .

8.73.101 SWFShape->drawCurveTo() : Dessine une courbe

`int swfshape->drawcurveto (int controlx , int controly , int anchorx , int anchory , int targetx , int targety)`

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[SWFShape->drawCurveTo](#) dessine une courbe quadratique (avec le style courant de ligne, modifié par [swfshape->setline](#)) depuis le point courant jusqu'au point (`anchorx` , `anchory`) en utilisant (`controlx` , `controly`) comme point de contrôle. C'est-à-dire qu'il commence en allant vers le point de contrôle, puis se dirige sur le point d'ancrage.

Avec 6 paramètres, la fonction dessine un cube de bézier du point ((`targetx` , `targety`) avec les points de contrôle (`controlx` , `controly`) et (`anchorx` , `anchory`).

Voir aussi [SWFShape->drawlineto](#) , [SWFShape->drawline](#) , [SWFShape->movepen](#) et [SWFShape->movepen](#) .

8.73.102 SWFShape::drawGlyph() : Dessine le premier caractère dans la chaîne de caractères donnée dans la forme en utilisant la définition de glyphe de la police donnée

`void SWFShape::drawGlyph (SWFFont font , string character , int size)`

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.103 SWFShape->drawLine() : Dessine une ligne relativement

`void swfshape->drawline (int dx , int dy)`

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[SWFShape->drawLine](#) dessine une ligne (avec le style courant de ligne, modifié par [swfshape->setline](#)) depuis le point courant, et sur le déplacement de (dx , dy).

Voir aussi [SWFShape->movepen](#) , [SWFShape->drawcurveto](#) , [SWFShape->movepen](#) et [SWFShape->drawlineto](#) .

8.73.104 SWFShape->drawLineTo() : Dessine une ligne

void **swfshape->drawlineto** (int x , int y)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[SWFShape->drawLineTo](#) dessine une ligne (avec le style courant de ligne, modifié par [swfshape->setline](#)) depuis le point courant jusqu'au point (x , y) dans l'espace de coordonnées de la forme.

Voir aussi [SWFShape->movepen](#) , [SWFShape->drawcurveto](#) , [SWFShape->movepen](#) et [SWFShape->drawline](#) .

8.73.105 SWFShape->movePen() : Déplace le stylo relativement

void **swfshape->movepen** (int dx , int dy)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[SWFShape->movePen](#) déplace le stylo dans la forme depuis les coordonnées (current x,current y) jusqu'au coordonnées (current x + dx , current y + dy), dans l'espace de coordonnées de la forme.

Voir aussi [SWFShape->movepen](#) , [SWFShape->drawcurveto](#) , [SWFShape->drawlineto](#) et [SWFShape->drawline](#) .

8.73.106 SWFShape->movePenTo() : Déplace le stylo

void **swfshape->movepen** (int x , int y)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[SWFShape->movePenTo](#) déplace le stylo dans la forme jusqu'au coordonnées globales (x , y).

Voir aussi [SWFShape->movepen](#) , [SWFShape->drawcurveto](#) , [SWFShape->drawlineto](#) et [SWFShape->drawline](#) .

8.73.107 SWFShape->setLeftFill() : Modifie la couleur de rasterisation de gauche

void **swfshape->setleftfill** (swfgradient fill)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

void **swfshape->setleftfill** (int red , int green , int blue , int a)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Tout ce sac de noeud fait qu'il y a deux couleurs de remplissage des lignes. Lorsque l'objet est rasterisé, il est pratique de savoir à l'avance quelle sont les remplissages, et le format SWF les demande.

[SWFShape->setLeftFill](#) affecte à la couleur de rasterisation de gauche, c'est-à-dire l'intérieur d'un objet, si vous définissez les contours d'un objet dans le sens inverse des aiguilles d'une montre. L'objet de remplissage est un objet [swffill](#) , retourné par la fonction [swfshape->addfill](#) ci-dessus.

Cela semble être le contraire lorsque vous définissez une forme dans un morphing. Si votre navigateur crashe, essayez de placer le remplissage sur l'autre côté.

Raccourci pour [SWFShape->setLeftFill\(\\$s->addfill\(\\$r, \\$g, \\$b \[, \\$a\]\)\);](#) .

Voir aussi [SWFShape->setLeftFill](#) .

8.73.108 SWFShape->setLine() : Modifie le style de ligne de la forme

void **swfshape->setline** (swfshape shape)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

void **swfshape->setline** (int width , int red , int green , int blue , int a)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[SWFShape->setLine](#) modifie le style de ligne de la forme. width est la largeur de la ligne. Si width vaut 0, le style est supprimé (et tous les autres arguments sont ignorés). Si width > 0, alors la couleur de la ligne devient (red , green , blue). Les couleurs sont représentées en RGB. Le dernier paramètre a est optionnel.

Vous devez déclarer un style avant de l'utiliser (voir exemple).

Cet exemple enfantin dessine une chaîne "!#%*@", dans des couleurs marrantes et un style rigolo.

Exemple `SWFShape->setLine`

```
<?php
$s = new SWFShape();
$f1 = $s->addFill(0xff, 0, 0);
$f2 = $s->addFill(0xff, 0x7f, 0);
$f3 = $s->addFill(0xff, 0xff, 0);
$f4 = $s->addFill(0, 0xff, 0);
$f5 = $s->addFill(0, 0, 0xff);
// erreur : il faut déclarer tous les styles avant
// de les utiliser.
$s->setLine(40, 0x7f, 0, 0);
$s->setLine(40, 0x7f, 0x3f, 0);
$s->setLine(40, 0x7f, 0x7f, 0);
$s->setLine(40, 0, 0x7f, 0);
$s->setLine(40, 0, 0, 0x7f);
$f = new SWFFont('Techno.fdb');
$s->setRightFill($f1);
$s->setLine(40, 0x7f, 0, 0);
$s->drawGlyph($f, '!');
$s->movePen($f->getWidth('!'), 0);
$s->setRightFill($f2);
$s->setLine(40, 0x7f, 0x3f, 0);
$s->drawGlyph($f, '#');
$s->movePen($f->getWidth('#'), 0);
$s->setRightFill($f3);
$s->setLine(40, 0x7f, 0x7f, 0);
$s->drawGlyph($f, '%');
$s->movePen($f->getWidth('%'), 0);
$s->setRightFill($f4);
$s->setLine(40, 0, 0x7f, 0);
$s->drawGlyph($f, '*');
$s->movePen($f->getWidth('*'), 0);
$s->setRightFill($f5);
$s->setLine(40, 0, 0, 0x7f);
$s->drawGlyph($f, '@');
$m = new SWFMovie();
$m->setDimension(3000,2000);
$m->setRate(12.0);
$i = $m->add($s);

// note la chaîne est ici!!!
$i->moveTo(1500-$f->getWidth("!#%*@")/2, 1000+$f->getAscent()/2);
header('Content-type: application/x-shockwave-flash');
$m->output();
?>
```

8.73.109 `SWFShape->setRightFill()` : Modifie la couleur de rastérioration de droite

void `swfshape->setrightfill` (swfgradient fill)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

void `swfshape->setrightfill` (int red , int green , int blue , int a)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Voir aussi [SWFShape->setleftfill](#) .

Raccourci pour `SWFShape->setRightFill($s->addfill($r, $g, $b [, $a]));` .

8.73.110 SWFShape() : Crée une nouvelle forme

SWFShape **swfshape** (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[SWFShape](#) crée une nouvelle forme.

SWFShape a les méthodes suivantes : [SWFShape->setline](#) , [SWFShape->addfill](#) , [SWFShape->setleftfill](#) , [SWFShape->setrightfill](#) , [SWFShape->movepeneto](#) , [SWFShape->movepen](#) , [SWFShape->drawlineto](#) , [SWFShape->drawline](#) , [SWFShape->drawcurveto](#) et [SWFShape->drawcurve](#) .

Ce exemple simple dessine un quadrant d'ellipse rouge.

Exemple avec [SWFShape](#)

```
<?php
$s = new SWFShape();
$s->setLine(40, 0x7f, 0, 0);
$s->setRightFill($s->addFill(0xff, 0, 0));
$s->movePenTo(200, 200);
$s->drawLineTo(6200, 200);
$s->drawLineTo(6200, 4600);
$s->drawCurveTo(200, 4600, 200, 200);
$m = new SWFMovie();
$m->setDimension(6400, 4800);
$m->setRate(12.0);
$m->add($s);
$m->nextFrame();
header('Content-type: application/x-shockwave-flash');
$m->output();
?>
```

8.73.111 SWFSound() : Retourne un nouvel objet SWFSound d'un fichier donné

SWFSound **SWFSound** (string filename , int flags)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.112 SWFSoundInstance::loopCount() :

void **SWFSoundInstance::loopCount** (int point)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.113 SWFSoundInstance::loopInPoint() :

void **SWFSoundInstance::loopInPoint** (int point)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.114 SWFSoundInstance::loopOutPoint() :

void **SWFSoundInstance::loopOutPoint** (int point)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.115 SWFSoundInstance::noMultiple() :

void **SWFSoundInstance::noMultiple** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.116 swfsprite->add() : Ajoute un objet à un sprite

void **swfsprite->add** (resource object)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

swfsprite->add ajoute une swfshape , un swfbutton , un swftext , une swfaction ou une autre animation swfsprite .

Pour les objets affichables (swfshape , swfbutton , swftext , swfaction or swfsprite), cela retourne une ressource sur l'objet dans la liste d'affichage.

8.73.117 SWFSprite::labelFrame() : Écrit un cadre

void **SWFSprite::labelFrame** (string label)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.118 SWFSprite->nextframe() : Va à la prochaine image du sprite

void **swfsprite->nextframe** (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFSprite->nextframe se déplace à la prochaine image du sprite.

8.73.119 SWFSprite->remove() : Supprime un objet dans un sprite

void **swfsprite->remove** (resource object)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFSprite->remove supprime une swfshape , un swfbutton , un swftext , une swfaction ou un SWFSprite du sprite courant.

8.73.120 SWFSprite->setframes() : Fixe le nombre maximum d'images dans le sprite

void **swfsprite->setframes** (int numberofframes)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFSprite->setframes fixe le nombre total d'images de l'animation à numberofframes .

8.73.121 SWFSprite::startSound() :

SWFSoundInstance **SWFSprite::startSound** (SWFSound sound)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.122 SWFSprite::stopSound() :

void **SWFSprite::stopSound** (SWFSound sound)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.123 SWFSprite() : Crée un sprite

SWFSprite **swfsprite** (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFSprite sont aussi connue sous le nom de "clip" : ils permettent la création d'objet animé dans une animation, avec un scénario propre. De ce fait, un sprite a les mêmes méthodes qu'une animation.

SWFSprite a les méthodes suivantes : SWFSprite->add , SWFSprite->remove , SWFSprite->nextframe et SWFSprite->setframes .

Ce exemple pratique fait tourner un superbe carré rouge.

Exemple de SWFSprite

```

<?php
$s = new SWFShape();
$s->setRightFill($s->addFill(0xff, 0, 0));
$s->movePenTo(-500,-500);
$s->drawLineTo(500,-500);
$s->drawLineTo(500,500);
$s->drawLineTo(-500,500);
$s->drawLineTo(-500,-500);
$p = new SWFSprite();
$i = $p->add($s);
$p->nextFrame();
$i->rotate(15);
$p->nextFrame();
$i->rotate(15);
$p->nextFrame();
$i->rotate(15);
$p->nextFrame();
$i->rotate(15);
$p->nextFrame();
$i->rotate(15);
$p->nextFrame();
$i->rotate(15);
$m = new SWFMovie();
$i = $m->add($p);
$i->moveTo(1500,1000);
$i->setName("blah");
$m->setBackground(0xff, 0xff, 0xff);
$m->setDimension(3000,2000);
header('Content-type: application/x-shockwave-flash');
$m->output();
?>

```

8.73.124 SWFText->addString() : Ajoute du texte

void **swftext->addstring** (string string)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`SWFText->addString` ajoute le texte string au texte courant, et le dessine. Le stylo est situé sur la ligne de base du texte, c'est-à-dire que le texte sera écrit horizontalement.

8.73.125 `SWFText::addUTF8String()` : Écrit le texte donné dans l'objet `SWFText` à la position du crayon, en utilisant la police, la hauteur, l'espacement et la couleur courante

`void SWFText::addUTF8String (string text)`

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.126 `SWFText::getAscent()` : Retourne la hampe de la police courante à sa taille courante ou 0 si elle n'est pas disponible

`float SWFText::getAscent (void)`

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.127 `SWFText::getDescent()` : Retourne le jambage de la police courante à sa taille courante ou 0 s'il n'est pas disponible

`float SWFText::getDescent (void)`

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.128 `SWFText::getLeading()` : Retourne l'interlignage de la police courante à sa taille courante ou 0 s'il n'est pas disponible

`float SWFText::getLeading (void)`

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.129 `SWFText::getUTF8Width()` : Calcule la largeur de la chaîne de caractères donnée dans les objets de texte de police et taille courante

`float SWFText::getUTF8Width (string string)`

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.130 `SWFText->getWidth()` : Calcule la longueur d'une chaîne

`float swftext->getWidth (string string)`

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFText->getWidth retourne la taille de la chaîne string , une fois qu'elle est dessinée avec la police et l'espacement courants.

8.73.131 SWFText->moveTo() : Déplace le stylo de texte

```
void swftext->moveto ( int x , int y )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFText->moveTo déplace le stylo (ou le curseur, si ça a un sens) jusqu'aux coordonnées (x , y) dans l'espace de coordonnées du texte. Si x ou y vaut 0, la valeur de coordonnées de la dimension reste la même. C'est ennuyeux, et cela devrait être corrigé.

8.73.132 SWFText->setColor() : Modifie la couleur de la police

```
void swftext->setcolor ( int red , int green , int blue , int a )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFText->setColor change la couleur de la police courante. Par défaut, c'est noir. La couleur est représentée avec la convention RGB.

8.73.133 SWFText->setFont() : Sélectionne la police courante

```
void swftext->setfont ( string font )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFText->setFont remplace la police courante par font .

8.73.134 SWFText->setHeight() : Modifie la hauteur de la police courante

```
void swftext->setheight ( int height )
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS

PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFText->setHeight fixe la hauteur courante de la police courante à `height` . Par défaut, c'est 240.

8.73.135 SWFText->setSpacing() : Modifie l'espacement de police

void **swftext->setspacing** (float spacing)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFText->setSpacing fixe l'espacement de police à `spacing` `spacing` . Par défaut, c'est 1.0. 0 signifie que toutes les lettres seront écrites au même point. Cela fonctionne pas terrible, car l'avance des lettres augmente, et l'espacement entre lettres n'est pas toujours le même. Il faudra que je l'explique plus clairement. Ou bien que je corrige les erreurs.

8.73.136 SWFText() : Crée un nouvel objet texte

SWFText **swftext** (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFText crée un nouvel objet texte, prêt à être manipulé.

SWFText a les méthodes suivantes : SWFText->setfont , SWFText->setheight , SWFText->setspacing , SWFText->setcolor , SWFText->moveto , SWFText->addstring et SWFText->getwidth .

Cet exemple simple va afficher la phrase " PHP fait du Flash avec Ming !! " sur un fond blanc.

Exemple avec SWFText

```
<?php
$f = new SWFFont("Techno.fdb");
$t = new SWFText();
$t->setFont($f);
$t->moveTo(200, 2400);
$t->setColor(0xff, 0xff, 0);
$t->setHeight(1200);
$t->addString("PHP fait du Flash avec Ming !!");
$m = new SWFMovie();
$m->setDimension(5400, 3600);
$m->add($t);
header('Content-type: application/x-shockwave-flash');
$m->output();
?>
```

8.73.137 SWFTextField::addChars() : Ajoute des caractères à une police qui seront disponibles dans un champ texte

void **SWFTextField::addChars** (string chars)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.138 SWFTextField->addstring() : Ajoute au texte

void **swftextfield->addstring** (string string)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
<u>SWFTextField->addstring</u> concatène la chaîne string avec la chaîne courante.

8.73.139 SWFTextField->align() : Modifie l'alignement du texte

void **swftextfield->align** (int alignement)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
<u>SWFTextField->align</u> change l'alignement du texte par alignement . Les valeurs valides pour alignement sont : SWFTEXTFIELD_ALIGN_LEFT , SWFTEXTFIELD_ALIGN_RIGHT , SWFTEXTFIELD_ALIGN_CENTER et SWFTEXTFIELD_ALIGN_JUSTIFY .

8.73.140 SWFTextField->setbounds() : Sélectionne la largeur et hauteur du champ

void **swftextfield->setbounds** (int width , int height)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
<u>SWFTextField->setbounds</u> fixe la longueur du champ à width et sa hauteur à height . Si vous ne fixez pas les bords vous-mêmes, Ming tentera de les deviner lui-même.

8.73.141 SWFTextField->setcolor() : Modifie la couleur du champ texte

void **swftextfield->setcolor** (int red , int green , int blue , int a)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
<u>SWFTextField->setcolor</u> modifie la couleur du champ texte, en la remplaçant par la couleur fournie. Par défaut, c'est noir opaque. Les couleurs sont représentées en convention RGB.

8.73.142 SWFTextField->setFont() : Modifie la police du champ

void **swftextfield->setfont** (string font)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFTextField->setFont remplace la police courante par la police font (police client ?).

8.73.143 SWFTextField->setHeight() : Modifie la hauteur de la police du champ texte

void **swftextfield->setheight** (int height)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFTextField->setHeight modifie la hauteur de la police du champ texte par height . Par défaut, c'est 240.

8.73.144 SWFTextField->setindentation() : Modifie l'indentation de la première ligne

void **swftextfield->setindentation** (int width)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFTextField->setindentation modifie l'indentation de la première ligne du champ texte, en la fixant à width .

8.73.145 SWFTextField->setLeftMargin() : Modifie la marge de gauche

void **swftextfield->setleftmargin** (int width)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFTextField->setLeftMargin modifie la marge de gauche du champ texte à width . Par défaut, c'est 0.

8.73.146 SWFTextField->setLineSpacing() : Modifie l'espacement de lignes

void **swftextfield->setlinespacing** (int height)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFTextField->setLineSpacing modifie l'espacement de lignes, en le fixant à height . Par défaut, c'est 40.

8.73.147 SWFTextField->setMargins() : Modifie les marges du champ texte Flash

void **swftextfield->setmargins** (int left , int right)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFTextField->setMargins modifie les deux marges du champ texte : left sera la nouvelle largeur de la marge de gauche et right , celle de droite. Par défaut, elles sont toutes les deux à 0.

8.73.148 SWFTextField->setname() : Nomme le champ texte

void **swftextfield->setname** (string name)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFTextField->setname baptise le champ texte name . Cela servira pour les formulaires et les actions.

8.73.149 SWFTextField::setPadding() : Fixe le remplissage d'un champ de texte

void **SWFTextField::setPadding** (float padding)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.150 SWFTextField->setrightMargin() : Modifie la marge de droite

void **swftextfield->setrightmargin** (int width)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, **SANS PREAVIS!** Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFTextField->setrightMargin modifie la marge de gauche du champ texte à width . Par défaut, c'est zéro.

8.73.151 SWFTextField() : Crée un nouveau champ texte

SWFTextField **swftextfield** (*int flags*)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, **SANS PREAVIS!** Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

SWFTextField crée un nouveau champ texte. Les champs textes sont moins souples que les swftext , car ils ne peuvent être tournés, mis à l'échelle ou inclinés, mais ils peuvent être utilisés sous forme de champs de formulaire, et ils peuvent utiliser des polices navigateur.

Les flags optionnels modifient les comportements du champ. Ils peuvent prendre les valeurs suivantes :

- SWFTEXTFIELD_DRAWBOX : dessine le contour du champ
- SWFTEXTFIELD_HASLENGTH
- SWFTEXTFIELD_HTML : autorise les tags HTML dans le texte
- SWFTEXTFIELD_MULTILINE : autorise les lignes multiples
- SWFTEXTFIELD_NOEDIT : indique que le champ ne peut pas être éditable.
- SWFTEXTFIELD_NOSELECT : le champ ne peut être sélectionné
- SWFTEXTFIELD_PASSWORD : indique que c'est un champ mot de passe
- SWFTEXTFIELD_WORDWRAP : autorise la mise en forme du texte

Les flags peuvent être combinés avec l'opérateur OR . Par exemple :

```
<?php
$t = newSWFTextField(SWFTEXTFIELD_PASSWORD | SWFTEXTFIELD_NOEDIT);
?>
```

crée un champ de mot de passe totalement inéritable (et inutile).

SWFTextField a les méthodes suivantes : SWFTextField->setfont , SWFTextField->setbounds , SWFTextField->align , SWFTextField->setheight , SWFTextField->setleftmargin , SWFTextField->setrightmargin , SWFTextField->setmargins , SWFTextField->setindentation , SWFTextField->setlinespacing , SWFTextField->setcolor , SWFTextField->setname et SWFTextField->addstring .

8.73.152 SWFVideoStream::getNumFrames() :

int **SWFVideoStream::getNumFrames** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.153 SWFVideoStream::setDimension() :

void **SWFVideoStream::setDimension** (int x , int y)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.73.154 SWFVideoStream() : Retourne un objet SWFVideoStream

SWFVideoStream **SWFVideoStream** (*string file*)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.74 Fonctions diverses

8.74.1 Introduction

Ces fonctions ont été placées là, car elles ne rentraient dans aucune catégorie adéquate.

8.74.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.74.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.74.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable	Historique
<code>ignore_user_abort</code>	"0"	PHP_INI_ALL	
<code>highlight.string</code>	"#DD0000"	PHP_INI_ALL	
<code>highlight.comment</code>	"#FF8000"	PHP_INI_ALL	
<code>highlight.keyword</code>	"#007700"	PHP_INI_ALL	
<code>highlight.bg</code>	"#FFFFFF"	PHP_INI_ALL	
<code>highlight.default</code>	"#0000BB"	PHP_INI_ALL	
<code>highlight.html</code>	"#000000"	PHP_INI_ALL	
<code>browscap</code>	NULL	PHP_INI_SYSTEM	

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du `php.ini`](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

`ignore_user_abort` bool

TRUE par défaut. Si changée à FALSE les scripts seront terminés aussitôt qu'ils essaieront d'envoyer des données à un client qui a annulé la connexion.

Voir aussi [ignore_user_abort](#).

`highlight.bg` string

`highlight.comment` string

`browscap` string

Nom du fichier de descriptif des clients HTML. (e.g.: `browscap.ini`) Voir aussi [get_browser](#).

8.74.5 Types de ressources

Cette extension ne définit aucune ressource.

8.74.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

CONNECTION_ABORTED ([entier](#))
CONNECTION_NORMAL ([entier](#))
CONNECTION_TIMEOUT ([entier](#))
__COMPILER_HALT_OFFSET__ ([entier](#))
Ajouté en PHP 5.1.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [connection_aborted](#)
- [connection_status](#)
- [connection_timeout](#)
- [constant](#)
- [define](#)
- [defined](#)
- [die](#)
- [eval](#)
- [exit](#)
- [get_browser](#)
- [__halt_compiler](#)
- [highlight_file](#)
- [highlight_string](#)
- [ignore_user_abort](#)
- [pack](#)
- [php_check_syntax](#)
- [php_strip_whitespace](#)
- [show_source](#)
- [sleep](#)
- [time_nanosleep](#)
- [time_sleep_until](#)
- [uniqid](#)
- [unpack](#)
- [usleep](#)

8.74.8 [connection_status\(\)](#) : Retourne les bits de statut de la connexion HTTP

int [connection_status](#) (void)

[connection_status](#) retourne les bits de statut de la connexion. Reportez-vous à la section [gestion des connexions](#) pour plus de détails.

Voir aussi [connection_aborted](#) et [ignore_user_abort](#) .

8.74.9 `connection_timeout()` : Indique si le script a expiré

bool `connection_timeout` (void)

`connection_timeout` retourne TRUE si le script a expiré. Reportez-vous à la section [gestion des connexions](#) pour plus de détails.

Attention

Déprécié

Cette fonction est dépréciée et n'existe plus depuis PHP 4.0.5.

Voyez la section sur la [Gestion de la connexion](#) dans la section [Caractéristiques](#) pour plus de détails.

Voir aussi [connection_status](#) .

8.74.10 `constant()` : Retourne la valeur d'une constante

mixed `constant` (string name)

`constant` retourne la valeur de la constante name .

`constant` est pratique lorsque vous devez lire la valeur d'une constante, mais que vous ne connaissez son nom que durant l'exécution du script. Par exemple, ce nom peut être le résultat d'une fonction.

Cette fonction fonctionne également avec les [constantes de classe](#) .

Exemple avec `constant`

```
<?php
  define ("MAXSIZE", 100);
  echo MAXSIZE;
  echo constant("MAXSIZE"); // identique a la ligne precedente
?>
```

Voir aussi [define](#) , [defined](#) et la section sur les [constantes](#) .

8.74.11 `define()` : Définit une constante

bool `define` (string name , mixed value , bool case_insensitive)

`define` définit la constante de nom name et de valeur value . Voyez la [section sur les constantes](#) pour plus de détails sur les constantes.

Le nom de la constante est donné par le paramètre name ; sa valeur est donnée par value .

Le troisième paramètre optionnel case_insensitive est une valeur booléenne. S'il vaut TRUE , le nom de la constante sera insensible à la casse : CONSTANT et Constant représentent des valeurs identiques. Par défaut, ces constantes représenteront des valeurs différentes.

Définition d'une constante

```

<?php
define ("CONSTANT", "Bonjour à tous.");
echo CONSTANT; // affiche "Bonjour à tous."
echo Constant; // affiche "Constant" et une alerte

define ("BONJOUR", "Salut toi.",TRUE);
echo BONJOUR; // affiche "Salut toi."
echo Bonjour; // affiche "Salut toi."

?>

```

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [defined](#) , [constant](#) et la section sur les [constantes](#) .

8.74.12 defined() : Vérifie l'existence d'une constante

bool **defined** (string name)

[defined](#) retourne TRUE si la constante nommée name a été définie, et FALSE sinon. Pour tester l'existence d'une fonction utilisez [function_exists](#) .

Vérifier la présence de constantes avec [defined](#)

```

<?php
if (defined("CONSTANT")){
// Notez que le nom de la constante est entre guillemets
    echo CONSTANT;
}
?>

```

Note

Si vous voulez vérifier si une variable existe, utilisez [isset](#) car [defined](#) ne s'applique qu'aux [constants](#) . Si vous voulez voir si une fonction existe, utilisez [function_exists](#) .

Voir aussi [define](#) , [constant](#) , [get_defined_constants](#) , la section sur les [constantes](#) et [function_exists](#) .

8.74.13 die() : Alias de la fonction exit

[die](#) est un [alias](#) de la fonction [exit](#) .

8.74.14 eval() : Exécute une chaîne comme un script PHP

mixed **eval** (string code_str)

[eval](#) évalue la chaîne code_str comme un script PHP. Parmi les utilisations possibles, cette fonction permet de stocker du code dans une base de données, pour utilisation ultérieure.

Il faut bien garder en tête que le code passé à [eval](#) doit être valide, y compris les points virgules de fin de lignes et les séquences d'échappement, sinon l'exécution se terminera.

N'oubliez pas que les variables utilisées dans la fonction [eval](#) resteront accessibles dans le script principal.

Une commande `return` terminera l'évaluation de la chaîne immédiatement. Depuis PHP 4, `eval` retourne `NULL` sauf si `return` est appelée dans le code évalué, auquel cas la valeur passée à `return` est retournée. Dans le cas d'une erreur de syntaxe dans le code évalué, `eval` retourne `FALSE`. Dans le cas d'une erreur fatale dans le code évalué, l'ensemble du script se terminera. En PHP 3, `eval` ne retourne pas de valeur.

Exemple avec `eval` - concaténation de texte

```
<?php
$string = 'tasse';
$name = 'café';
$str = 'Ceci est une $string avec mon $name dedans.<br />';
echo $str;
eval( "\$str = \"\$str\";" );
echo $str;
?>
```

L'exemple ci-dessus va afficher :

```
Ceci est une $string avec mon $name dedans.
Ceci est une tasse avec mon café dedans.
```

Array

Voir aussi [call_user_func](#).

8.74.15 `exit()` : Termine le script courant

`void exit (string status) void exit (int status)`

Note

`exit` n'est pas une véritable fonction, mais un élément de langage.

Note

PHP >= 4.2.0 n'affiche pas le message status si ce n'est pas un entier entier.

`exit` termine l'exécution du script courant. Elle n'a pas de valeur de retour (et pour cause !), mais elle utilisera le message status comme message de fin d'exécution.

Si le paramètre status est un entier, cette valeur pourra être utilisée comme statut de sortie. Le statut de sortie peut être dans l'intervalle 0-255, le statut de sortie 255 est réservé par PHP et ne doit pas être utilisé. Le statut 0 est utilisé pour terminer le programme avec succès.

Exemple avec `exit`

```
<?php
$filename = '/chemin/jusqua/fichier';
$file = fopen ($filename, 'r')
    or exit("Impossible d'ouvrir le fichier $filename");
?>
```

Exemple avec `exit` et un code d'erreur

```
<?php
//quitte normalement
exit;
exit();
```

```

exit(0);

//quitte avec un code d'erreur
exit(1);
exit(0376); //octal

?>

```

Note

La fonction die est un alias de la fonction exit .

Voir aussi register_shutdown_function .

8.74.16 get_browser() : Indique de quoi est capable le navigateur client

mixed **get_browser** (*string user_agent* , *bool return_array*)

get_browser essaie de déterminer les capacités du navigateur client. Cela se fait en lisant les informations dans le fichier browscap.ini .

Par défaut, la valeur de HTTP User-Agent est utilisée. Cependant, vous pouvez passer n'importe quelle valeur avec le paramètre optionnel *user_agent* à get_browser . Vous pouvez omettre *user_agent* en passant la valeur NULL .

Les informations sont retournées sous forme d'un objet, dont les différents membres contiendront des informations, telles que les versions majeures et mineures et des chaînes d'identification ; des booléens pour des caractéristiques telles que frames, JavaScript, et cookies ; et ainsi de suite.

Depuis PHP 4.3.2, si le paramètre optionnel *return_array* vaut TRUE , cette fonction retournera un tableau au lieu d'un objet .

Même si browscap.ini contient des informations sur de nombreux clients, il compte sur les utilisateurs pour être mis à jour. Le format du fichier est facilement compréhensible.

L'exemple suivant montre comment on peut lister les informations disponibles :

Exemple avec get_browser : liste des capacités du navigateur du client

```

<?php
echo $_SERVER['HTTP_USER_AGENT'] . "\n\n";

$browser = get_browser(null, true);
print_r($browser);

?>

```

L'exemple ci-dessus va afficher quelque chose de similaire à :

Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7) Gecko/20040803 Firefox/0.9.3

Array

```

(
[browser_name_regex] => ^mozilla/5\.0 (windows; .; windows nt 5\.1; .*rv:.* ) gecko/. * firefox/0\.9.*$
[browser_name_pattern] => Mozilla/5.0 (Windows; ?; Windows NT 5.1; *rv:*) Gecko/* Firefox/0.9*
[parent] => Firefox 0.9
[platform] => WinXP

```

```
[browser] => Firefox
[version] => 0.9
[majorver] => 0
[minorver] => 9
[css] => 2
[frames] => 1
[iframes] => 1
[tables] => 1
[cookies] => 1
[backgroundsounds] =>
[vbscript] =>
[javascript] => 1
[javaapplets] => 1
[activexcontrols] =>
[cdf] =>
[aol] =>
[beta] => 1
[win16] =>
[crawler] =>
[stripper] =>
[wap] =>
[netclr] =>
)
```

La valeur cookies indique simplement que le navigateur est capable d'accepter les cookies, et n'indique pas si l'utilisateur les a activé sur son navigateur. Le seul moyen de tester l'activation des cookies est d'en poser un avec la fonction [setcookie](#), de recharger la page et de vérifier que le cookie existe toujours.

Note

Afin de pouvoir fonctionner, la directive de configuration [browscap](#) dans le fichier php.ini doit pointer vers le fichier browscap.ini de votre système.

browscap.ini n'est pas distribué avec PHP, mais vous pouvez le télécharger sur [up-to-date php browscap.ini](#).

Bien que browscap.ini contienne des informations sur de très nombreux navigateurs, il incombe à l'utilisateur de conserver sa base de données à jour. Le format du fichier est très simple à comprendre.

8.74.17 `__halt_compiler()` : Stoppe l'exécution du compilateur

```
void __halt_compiler ( void )
```

Cette fonction stoppe l'exécution du compilateur. Ceci peut être très utile pour embarquer des données dans des scripts PHP, comme des fichiers d'installation. L'octet de la position du début des données peut être déterminé par la constante `__COMPILER_HALT_OFFSET__` qui n'est définie que s'il y a une fonction `__halt_compiler` présente dans le fichier.

Note

`__halt_compiler` ne peut être utilisé que depuis une portée extérieure.

8.74.18 highlight_file() : Colorisation syntaxique d'un fichier

mixed **highlight_file** (string filename , bool return)

highlight_file affiche la syntaxe colorisée du fichier filename , en utilisant les couleurs définies dans le moteur interne de PHP.

Si le second paramètre optionnel return vaut TRUE alors highlight_file retournera le code généré, au lieu de l'afficher. Si le second paramètre ne vaut pas TRUE alors highlight_file retournera TRUE en cas de succès, et FALSE sinon.

Note
Le paramètre return a été ajouté en PHP 4.2.0. Avant cela, la fonction se comportait comme si ce paramètre avait sa valeur par défaut à FALSE .
Note
Depuis PHP 4.2.1, cette fonction est aussi affectée par <u>safe_mode</u> et <u>open_basedir</u> .
Attention
Beaucoup de soin doit être apporté lors de l'utilisation de <u>show_source</u> et <u>highlight_file</u> pour s'assurer que vous ne révélez pas d'informations critiques telles que des mots de passe ou d'autres informations qui pourraient créer des fuites d'information.
De nombreux serveurs sont configurés pour automatiquement afficher le source colorisé, avec l'extension phps . Par exemple, exemple.phps va afficher le source du script. Pour activer cette fonctionnalité, utilisez cette ligne dans httpd.conf :

```
AddType application/x-httpd-php-source .phps
```

Voir aussi highlight_string .

8.74.19 highlight_string() : Applique la syntaxe colorisée à du code PHP

mixed **highlight_string** (string str , bool return)

highlight_string affiche la version colorisée du code PHP contenu dans le paramètre str , en utilisant les couleurs du système interne de colorisation de PHP.

Si le second paramètre optionnel return est fourni, et vaut TRUE alors highlight_string retournera la chaîne colorisée au lieu de l'afficher immédiatement. Si le second paramètre ne vaut pas TRUE alors highlight_string retournera TRUE en cas de succès, et FALSE sinon.

Exemple avec <u>highlight_string</u>
<pre><?php highlight_string('<?php phpinfo(); ?>'); ?></pre>
Cet exemple affichera (en PHP 4):
<pre><code> <?php phpinfo()); ?></pre>

```
</font>
</code>
```

Cet exemple affichera (en PHP 5):

```
<code><span style="color: #000000">
<span style="color: #0000BB"><?php phpinfo</span><span style="color: #007700">());
</span><span style="color: #0000BB">?></span>
</span>
</code>
```

Note

Le paramètre `return` est disponible depuis PHP 4.2.0. Auparavant, il prenait la valeur par défaut de `FALSE`

Voir aussi [highlight_file](#) et [show_source](#) .

8.74.20 `ignore_user_abort()` : Active l'interruption de script sur déconnexion du visiteur

```
int ignore_user_abort ( bool setting )
```

`ignore_user_abort` active l'option décidant si, lors de la déconnexion du client, le script doit poursuivre son exécution ou non. La fonction renvoie le paramétrage précédent et elle peut être appelée sans argument pour ne pas changer le paramétrage courant. `setting` est un booléen, qui active ou désactive l'interruption. Voir le paragraphe sur la [gestion des connexions](#) dans le chapitre [fonctionnalités](#) pour une description plus complète des manipulations de connexion avec PHP.

Voir aussi [connection_aborted](#) et [connection_status](#) .

8.74.21 `pack()` : Compacte des données dans une chaîne binaire

```
string pack ( string format , mixed args , mixed ... )
```

`pack` compacte les arguments `args` dans une chaîne binaire, suivant le format `format` . `pack` retourne la chaîne binaire.

L'idée vient du Perl et tout le formatage fonctionne de la même façon qu'en Perl, mais quelques formats manquent encore (comme "u"). La chaîne de format est composée d'une série de codes de formats, suivis par un quantificateur optionnel. Le quantificateur peut être un entier ou * pour la répétition indéfinie. Pour les formats a, A, h et H, le quantificateur spécifie combien de caractères d'un argument sont pris; pour @, c'est la position absolue où placer les données, et pour le reste, c'est le nombre de répétitions. Actuellement, les formats suivants sont implémentés.

Code	Description
a	Une chaîne complétée avec NULL
A	Une chaîne complétée avec espace (SPACE)
h	Chaîne hexadécimale h, bit de poids faible en premier.
H	Chaîne hexadécimale H, bit de poids fort en premier.
c	caractère signé
C	caractère non signé

s	entier court signé (toujours sur 16 bits, ordre des bits dépendant de la machine).
S	entier court non signé (toujours 16 bits, ordre des bits dépendant de la machine).
n	entier court non signé (toujours 16 bits, ordre des bits big endian)
v	entier court non signé (toujours 16 bits, ordre des bits little endian)
i	entier signé (taille et ordre des bits dépendants de la machine)
l	entier non signé (taille et ordre des bits dépendants de la machine)
l	entier long signé (toujours 32 bits, ordre des bits dépendant de la machine)
L	entier long non signé (toujours 32 bits, ordre des bits dépendant de la machine)
N	entier long non signé (toujours 32 bits, ordre des bits big endian)
V	entier long non signé (toujours 32 bits, ordre des bits little endian)
f	nombre à virgule flottante (taille et représentation dépendantes de la machine)
d	nombre à virgule flottante double (taille et représentation dépendantes de la machine)
x	caractère NUL
X	Reculé d'un caractère
@	Remplit avec des NUL jusqu'à la position absolue

Compactage d'une chaîne
<pre><?php \$binarydata = pack ("nvc*", 0x1234, 0x5678, 65, 66); ?></pre> <p>La chaîne binaire résultante aura 6 octets de long, et contiendra la séquence 0x12, 0x34, 0x78, 0x56, 0x41, 0x42.</p>

Notez que la distinction entre signé et non signé n'affecte que la fonction unpack, tandis que la fonction pack fournira le même résultat pour les deux formats.

De plus, notez que PHP enregistre de manière interne et intégrale les valeurs : cette représentation dépend de la machine. Si vous essayez d'enregistrer une valeur trop grande, elle risque d'être convertie et de donner lieu à des effets de bords vicieux.

Voir aussi unpack .

8.74.22 php_check_syntax() : Vérifie la syntaxe PHP (et exécute) d'un fichier spécifique

bool php_check_syntax (string file_name , string error_message)

Pour des raisons techniques, cette fonction est dépréciée et supprimée de PHP. Utilisez à la place `php -l somefile.php` depuis la ligne de commande .

php_check_syntax effectue une vérification de la syntaxe du fichier filename pour y trouver des erreurs de script. Ceci est similaire à utiliser la commande `php -l` depuis la ligne de commande mise à part que php_check_syntax exécutera (mais n'affichera rien) le fichier file_name vérifié. Par exemple, si une fonction est définie dans file_name , cette fonction sera disponible pour le fichier qui exécute php_check_syntax , mais l'affichage depuis file_name sera supprimée.

8.74.23 `php_strip_whitespace()` : Retourne la source sans commentaires ni espaces blancs

string `php_strip_whitespace` (string filename)

`php_strip_whitespace` retourne le code source PHP du paramètre filename en ayant supprimé les commentaires ainsi que les espaces. Cela peut être utile pour comparer la quantité de code avec la quantité de commentaire dans votre code. Cela revient à utiliser la commande `php -w` depuis la [ligne de commande](#) .

Note

Cette fonction fonctionne comme décrite depuis PHP 5.0.1. Avant, elle ne retournait qu'une chaîne vide. Pour plus de détails concernant ce bogue ainsi que son comportement précédent, lisez le [rapport de bogue #29606](#) .

8.74.24 `show_source()` : Alias de `highlight_file`

Cette fonction est un alias de : [highlight_file](#) .

8.74.25 `sleep()` : Arrête l'exécution durant quelques secondes

int `sleep` (int seconds)

`sleep` retarde l'exécution du programme pendant seconds secondes.

Exemple avec `sleep`

```
<?php
// Heure actuelle
echo date('h:i:s') . "\n";

// stoppe pour 10 secondes
sleep(10);

// retour!
echo date('h:i:s') . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
05:31:23
05:31:33
```

Voir aussi [usleep](#) et [set_time_limit](#)

8.74.26 `time_nanosleep()` : Attendre pendant un nombre de secondes et de nanosecondes

mixed `time_nanosleep` (int seconds , int nanoseconds)

`time_nanosleep` permet d'imposer un délai d'exécution à un programme pendant un nombre de secondes `seconds` et un nombre de nanosecondes `nanoseconds` .

`seconds` doit être un entier positif et `nanoseconds` doit être un entier positif plus petit que 1 billion.

8.74.27 `time_sleep_until()` : Stoppe le script pendant une durée spécifiée

`bool time_sleep_until (float timestamp)`

Cette fonction stoppe le script pendant la durée spécifiée `timestamp` . Si la durée `timestamp` est dans le passé, `time_sleep_until` générera une alerte de niveau `E_WARNING` et retournera `FALSE` .

Note

Cette fonction n'est pas implémentée sous Windows.
--

8.74.28 `uniqid()` : Génère un identifiant unique

`string uniqid (string prefix , bool more_entropy)`

`uniqid` retourne un identifiant préfixé unique, basé sur l'heure courante, en micro-secondes. Le paramètre `prefix` est optionnel mais peut servir à identifier facilement différents hôtes, si vous générez simultanément des fichiers depuis plusieurs hôtes, à la même micro-seconde. Depuis PHP 4.3.1, `prefix` peut prendre jusqu'à 114 caractères.

Si le paramètre optionnel `more_entropy` est `TRUE` , `uniqid` ajoutera une entropie "combined LCG" à la fin de la valeur retournée, ce qui renforcera encore l'unicité de l'identifiant.

Sans `prefix` (préfixe vide), la chaîne retournée fera 13 caractères. Si `more_entropy` est à `TRUE` , elle fera 23 caractères.

Note

Le paramètre <code>prefix</code> est devenu optionnel en PHP 5.

Si vous voulez utiliser un identifiant unique, ou bien gérer des cookies, il est recommandé d'utiliser un code tel que celui-ci :

Exemple avec <code>uniqid</code>

<pre><?php // Aucun préfixe \$token = md5 (uniqid ()); // meilleur, difficile à deviner \$better_token = md5(uniqid(rand(), true)); ?></pre>

Ceci va créer un identifiant de 32 caractères (un nombre hexadécimal de 128) qui sera très difficile à prédire.

8.74.29 `unpack()` : Déconditionne des données depuis une chaîne binaire

array **unpack** (string format , string data)

unpack déconditionne les données data depuis une chaîne binaire avec le format format . unpack retourne un tableau contenant les éléments déconditionnés.

unpack se comporte légèrement différemment de la version Perl car les données déconditionnées sont stockées dans un tableau. Pour cela, il faut donner un nom à chaque format utilisé et les séparer par des slash (/).

Exemple avec unpack

```
<?php
    $array = unpack ("c2chars/nint", $binarydata);
?>
```

Le tableau résultant contiendra les entrées suivantes : "chars1", "chars2" et "int".

Pour plus de détails, reportez-vous à : pack

Attention

Il faut noter que PHP gère les valeurs en interne sous forme signée. Si vous déconditionnez une valeur qui est aussi grande que la taille utilisée en interne par PHP, le résultat se trouvera être un nombre négatif, même s'il a été déconditionné avec l'option " non signé ".

Voir aussi pack pour des informations sur les formats disponibles.

8.74.30 usleep() : Retarde l'exécution en micro-secondes

void **usleep** (int micro_seconds)

sleep retarde l'exécution du programme pendant micro_seconds micro-secondes.

Exemple avec sleep

```
<?php

// Heure courante
echo date('h:i:s') . "\n";

// stoppe pour 2 secondes
usleep(2000000);

// retour!
echo date('h:i:s') . "\n";

?>
```

L'exemple ci-dessus va afficher :

```
11:13:28
11:13:30
```

Note

usleep était inopérante sous Windows jusqu'en version PHP 5.0.0.

Voir aussi sleep .

8.75 mnoGoSearch

8.75.1 Introduction

Ces fonctions donnent l'accès à mnoGoSearch (anciennement UdmSearch), moteur de recherche du monde libre. mnoGoSearch est un moteur de recherche complet, destiné aux intranets et serveurs web, distribué sous licence GNU. MnoGoSearch offre des fonctionnalités unique, qui en font un excellent outil pour un grand nombre d'applications de recherche dans votre site : recherche de recettes de cuisines ou dans les journaux, recherche dans un site FTP, dans les groupes de news, etc. Il offre un système d'indexation de textes pour les fichiers HTML, PDF et documents textes. MnoGoSearch est constitué de deux parties : l'indexeur, qui effectue les recherches et le moteur de recherche. L'indexeur passe en revue récursivement les sites HTTP, FTP, NEWS ou encore les fichiers locaux, et enregistre des méta-données dans les bases MySQL, pour optimiser les recherches ultérieures. Une fois que tous les documents ont été référencés, ils sont accessibles au moteur de recherche. Celui-ci est utilisable par interface web. Les langages C CGI, Perl et PHP sont supportés pour effectuer les recherches.

Plus de détails sur le site officiel de mnoGoSearch : <http://www.mnogosearch.org/> .

Note
Cette extension a été déplacée dans le module <u>PECL</u> et ne sera plus intégrée dans PHP à partir de PHP 5.1.0.
Note
Cette extension n'est pas disponible sur les plates-formes Windows.

8.75.2 Pré-requis

Téléchargez mnoGosearch sur le site <http://www.mnogosearch.org/> et installez-le sur votre système. Vous aurez besoin de la version 3.1.10 ou plus récente de mnoGoSearch, pour pouvoir utiliser ces fonctions.

8.75.3 Installation

Pour pouvoir les utiliser, vous devez inclure le support en ajoutant l'option `--with-mnogosearch` . Si vous utilisez cette option sans indiquer le chemin jusqu'à mnogosearch, PHP essaiera de le trouver dans le dossier `/usr/local/mnogosearch` . Si vous avez installé mnogosearch dans un autre endroit, vous devez l'indiquer comme ceci : `--with-mnogosearch=DIR` .

Note
PHP supporte naturellement MySQL. Il faut savoir que mnoGoSearch n'est pas compatible avec la bibliothèque interne de PHP, et ne peut fonctionner qu'avec les bibliothèques génériques MySQL. Par conséquent, si vous utilisez mnoGoSearch avec MySQL, indiquez le dossier d'installation de MySQL durant la configuration avec l'option : <code>--with-mnogosearch --with-mysql=/usr</code> .

8.75.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.75.5 Types de ressources

8.75.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

UDM_FIELD_URLID (entier)
 UDM_FIELD_URL (entier)
 UDM_FIELD_CONTENT (entier)
 UDM_FIELD_TITLE (entier)
 UDM_FIELD_KEYWORDS (entier)
 UDM_FIELD_DESC (entier)
 UDM_FIELD_DESCRIPTION (entier)
 UDM_FIELD_TEXT (entier)
 UDM_FIELD_SIZE (entier)
 UDM_FIELD_RATING (entier)
 UDM_FIELD_SCORE (entier)
 UDM_FIELD_MODIFIED (entier)
 UDM_FIELD_ORDER (entier)
 UDM_FIELD_CRC (entier)
 UDM_FIELD_CATEGORY (entier)
 UDM_FIELD_LANG (entier)
 UDM_FIELD_CHARSET (entier)
 UDM_PARAM_PAGE_SIZE (entier)
 UDM_PARAM_PAGE_NUM (entier)
 UDM_PARAM_SEARCH_MODE (entier)
 UDM_PARAM_CACHE_MODE (entier)
 UDM_PARAM_TRACK_MODE (entier)
 UDM_PARAM_PHRASE_MODE (entier)
 UDM_PARAM_CHARSET (entier)
 UDM_PARAM_LOCAL_CHARSET (entier)
 UDM_PARAM_BROWSER_CHARSET (entier)
 UDM_PARAM_STOPTABLE (entier)
 UDM_PARAM_STOP_TABLE (entier)
 UDM_PARAM_STOPFILE (entier)
 UDM_PARAM_STOP_FILE (entier)
 UDM_PARAM_WEIGHT_FACTOR (entier)
 UDM_PARAM_WORD_MATCH (entier)
 UDM_PARAM_MAX_WORD_LEN (entier)
 UDM_PARAM_MAX_WORDLEN (entier)
 UDM_PARAM_MIN_WORD_LEN (entier)
 UDM_PARAM_MIN_WORDLEN (entier)
 UDM_PARAM_ISPELL_PREFIXES (entier)
 UDM_PARAM_ISPELL_PREFIX (entier)
 UDM_PARAM_PREFIXES (entier)
 UDM_PARAM_PREFIX (entier)
 UDM_PARAM_CROSS_WORDS (entier)
 UDM_PARAM_CROSSWORDS (entier)
 UDM_PARAM_VARDIR (entier)
 UDM_PARAM_DATADIR (entier)
 UDM_PARAM_HLBEG (entier)
 UDM_PARAM_HLEND (entier)

UDM_PARAM_SYNONYM ([entier](#))
UDM_PARAM_SEARCHD ([entier](#))
UDM_PARAM_QSTRING ([entier](#))
UDM_PARAM_REMOTE_ADDR ([entier](#))
UDM_LIMIT_CAT ([entier](#))
UDM_LIMIT_URL ([entier](#))
UDM_LIMIT_TAG ([entier](#))
UDM_LIMIT_LANG ([entier](#))
UDM_LIMIT_DATE ([entier](#))
UDM_PARAM_FOUND ([entier](#))
UDM_PARAM_NUM_ROWS ([entier](#))
UDM_PARAM_WORDINFO ([entier](#))
UDM_PARAM_WORD_INFO ([entier](#))
UDM_PARAM_SEARCHTIME ([entier](#))
UDM_PARAM_SEARCH_TIME ([entier](#))
UDM_PARAM_FIRST_DOC ([entier](#))
UDM_PARAM_LAST_DOC ([entier](#))
UDM_MODE_ALL ([entier](#))
UDM_MODE_ANY ([entier](#))
UDM_MODE_BOOL ([entier](#))
UDM_MODE_PHRASE ([entier](#))
UDM_CACHE_ENABLED ([entier](#))
UDM_CACHE_DISABLED ([entier](#))
UDM_TRACK_ENABLED ([entier](#))
UDM_TRACK_DISABLED ([entier](#))
UDM_PHRASE_ENABLED ([entier](#))
UDM_PHRASE_DISABLED ([entier](#))
UDM_CROSS_WORDS_ENABLED ([entier](#))
UDM_CROSSWORDS_ENABLED ([entier](#))
UDM_CROSS_WORDS_DISABLED ([entier](#))
UDM_CROSSWORDS_DISABLED ([entier](#))
UDM_PREFIXES_ENABLED ([entier](#))
UDM_PREFIX_ENABLED ([entier](#))
UDM_ISPELL_PREFIXES_ENABLED ([entier](#))
UDM_ISPELL_PREFIX_ENABLED ([entier](#))
UDM_PREFIXES_DISABLED ([entier](#))
UDM_PREFIX_DISABLED ([entier](#))
UDM_ISPELL_PREFIXES_DISABLED ([entier](#))
UDM_ISPELL_PREFIX_DISABLED ([entier](#))
UDM_ISPELL_TYPE_AFFIX ([entier](#))
UDM_ISPELL_TYPE_SPELL ([entier](#))
UDM_ISPELL_TYPE_DB ([entier](#))
UDM_ISPELL_TYPE_SERVER ([entier](#))
UDM_MATCH_WORD ([entier](#))
UDM_MATCH_BEGIN ([entier](#))
UDM_MATCH_SUBSTR ([entier](#))
UDM_MATCH_END ([entier](#))

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)

- [udm_add_search_limit](#)
- [udm_alloc_agent_array](#)
- [udm_alloc_agent](#)
- [udm_api_version](#)
- [udm_cat_list](#)
- [udm_cat_path](#)
- [udm_check_charset](#)
- [udm_check_stored](#)
- [udm_clear_search_limits](#)
- [udm_close_stored](#)
- [udm_crc32](#)
- [udm_errno](#)
- [udm_error](#)
- [udm_find](#)
- [udm_free_agent](#)
- [udm_free_ispell_data](#)
- [udm_free_res](#)
- [udm_get_doc_count](#)
- [udm_get_res_field](#)
- [udm_get_res_param](#)
- [udm_hash32](#)
- [udm_load_ispell_data](#)
- [udm_open_stored](#)
- [udm_set_agent_param](#)

8.75.8 [udm_alloc_agent_array\(\)](#) : Alloue une session mnoGoSearch

resource [udm_alloc_agent_array](#) (array databases)

[udm_alloc_agent_array](#) sert à créer un agent avec des connexions à de multiples bases de données. Le paramètre `databases` est un tableau qui contient une URL par élément, de manière analogue au premier paramètre de [udm_alloc_agent](#).

Voir aussi [udm_alloc_agent](#).

8.75.9 [udm_alloc_agent\(\)](#) : Alloue une session mnoGoSearch

resource [udm_alloc_agent](#) (string dbaddr , string dbmode)

[udm_alloc_agent](#) retourne un agent mnogosearch en cas de succès, FALSE en cas d'erreur. [udm_alloc_agent](#) crée une session avec les paramètres de base de données.

`dbaddr` est une description de base de données formatée comme une URL. Les options (type, hôte, nom de base de données, port, utilisateur ou mot de passe) servent à se connecter à la base de données SQL. Ne passez aucune valeur si vous souhaitez utiliser le support des fichiers texte intégré. Sinon, utilisez le format : `DBAddr DBType:[//[DBUser[:DBPass]@]DBHost[:DBPort]]/DBName/`. Actuellement, les valeurs de DBType possibles sont : `mysql`, `pgsql`, `msql`, `solid`, `mssql`, `oracle`, `ibase`. En fait, si vous avez ajouté un support natif, cette option est inutile. Mais les utilisateurs ODBC doivent spécifier une des valeurs supportées. Si votre type de base de données n'est pas supporté, utilisez le terme "unknown".

dbmode - Vous pouvez sélectionner le mode de stockage des mots dans la base de données. Si vous indiquez "single", tous les mots seront stockés dans la même table. Si vous indiquez "multi", ils seront situés dans différentes tables, suivant leur taille. Le mode "multi" est généralement plus rapide, mais requiert plus de tables. Si le mode "crc" est sélectionné, mnoGoSearch enregistrera un entier de 32 bits, calculé avec l'algorithme CRC32, plutôt que des mots. Ce mode requiert moins d'espace disque, et il est beaucoup plus rapide que les modes "single" et "multi". "crc-multi" utilise la même technique de stockage que le mode "crc", mais il stocke aussi les mots dans différentes tables suivant leur taille. Format: DBMode single/multi/crc/crc-multi.

Note
dbaddr et dbmode doit correspondre à ceux qui sont utilisés lors de l'indexation.
Note
En réalité, <u>udm_alloc_agent</u> n'ouvre pas de connexion, et donc, ne vérifie ni le nom d'utilisateur, ni le mot de passe. La connexion à une base de données et les vérifications des login / mot de passe sont réalisées à l'appel de la fonction <u>udm_find</u> .

8.75.10 udm_api_version() : Lit la version de l'API mnoGoSearch

int udm_api_version (void)

udm_api_version retourne le numéro de version de l'API mnoGoSearch. Par exemple, si mnoGoSearch 3.1.10 est utilisé, udm_api_version retournera 30110 .

udm_api_version permet aux utilisateurs d'identifier quelles sont les API disponibles. Par exemple, udm_get_doc_count n'est disponible qu'à partir de mnoGoSearch 3.1.11.

Exemple avec <u>udm_api_version</u>
<pre><?php if (udm_api_version() >= 30111) { echo "Nombre total d'URL dans la base : " . udm_get_doc_count(\$udm) . "
\n"; } ?></pre>

8.75.11 udm_cat_list() : Liste toutes les catégories soeurs d'une catégorie

array udm_cat_list (resource agent , string category)

udm_cat_list retourne un tableau contenant la liste de toutes les catégories de même niveau que la catégorie category courante. agent est l'identifiant de l'agent retourné par un appel à la fonction udm_alloc_agent .

Cette fonction est pratique pour réaliser des arbres à partir des catégories.

udm_cat_list retourne un tableau avec le format suivant :

Le tableau est constitué de paires. Les index pairs contiennent les chemins de catégories, les index impairs les noms des catégories correspondantes.

```
$array[0] contiendra '020300'
$array[1] contiendra 'Marseille'
```

\$array[2] contiendra '020301'
 \$array[3] contiendra 'Lille'
 \$array[4] contiendra '020302'
 \$array[5] contiendra 'Lyon'
 ...
 etc.

L'exemple ci-dessus va afficher :

Marseille
 Lille
 Lyon
 ...

<p>Exemple avec <code>udm_cat_list</code></p> <pre> <?php \$cat_list_arr = udm_cat_list(\$udm_agent,\$cat); \$cat_list = ''; for (\$i=0; \$i<count(\$cat_list_arr); \$i+=2) { \$path = \$cat_list_arr[\$i]; \$name = \$cat_list_arr[\$i+1]; \$cat_list .= '' . \$name . '
'; } ?> </pre>

Voir aussi [udm_cat_path](#) .

8.75.12 `udm_cat_path()` : Lit le chemin de la catégorie courante

array `udm_cat_path` (resource agent , string category)

`udm_cat_path` retourne un tableau listant les catégories depuis la racine jusqu'à la catégorie courante. Le paramètre agent est un identifiant de résultat, obtenu après un appel à `udm_alloc_agent` .

category - La catégorie courante : celle dont on veut le chemin.

`udm_cat_path` retourne un tableau avec le format suivant :

Le tableau est constitué de paires. Les index pairs contiennent les chemins de catégories, les index impairs les noms des catégories correspondantes.

Par exemple, l'appel `$array=udm_cat_path($agent, '02031D');` peut retourner le tableau suivant :

\$array[0] contiendra "
 \$array[1] contiendra 'Root'
 \$array[2] contiendra '02'
 \$array[3] contiendra 'Sport'
 \$array[4] contiendra '0203'
 \$array[5] contiendra 'Foot'
 \$array[4] contiendra '02031D'
 \$array[5] contiendra 'PSG'

Spécifier une catégorie avec un format hiérarchisé

```

<?php
//Spécifier le chemin de la catégorie courante avec le format suivant
// '> Root > Sport > Foot > PSG'
$cat_path_arr = udm_cat_path($udm_agent, $cat);
$cat_path = '';
for ($i=0; $i<count($cat_path_arr); $i+=2) {
    $path = $cat_path_arr[$i];
    $name = $cat_path_arr[$i+1];
    $cat_path .= " > <a href=\"$_SERVER[PHP_SELF]?cat=$path\">$name</a> ";
}
?>

```

Voir aussi [udm_cat_list](#) .

8.75.13 udm_check_charset() : Vérifie si un jeu de caractères est supporté par mnogosearch

bool **udm_check_charset** (resource agent , string charset)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.75.14 udm_check_stored() : Vérifie la connexion MnoGoSearch avec le document stocké

int **udm_check_stored** (resource agent , int link , string doc_id)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.75.15 udm_clear_search_limits() : Annule toutes les limitations de recherche MnoGoSearch

bool **udm_clear_search_limits** (resource agent)

[udm_clear_search_limits](#) annule toutes les limitations de recherche imposées, et retourne TRUE .

Voir aussi [udm_add_search_limit](#) .

8.75.16 udm_close_stored() : Ferme la connexion MnoGoSearch avec le document enregistré

int **udm_close_stored** (resource agent , int link)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.75.17 `udm_crc32()` : Retourne la somme de contrôle CRC32 d'une chaîne

int `udm_crc32` (resource agent , string str)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.75.18 `udm_errno()` : Numéro d'erreur mnoGoSearch

int `udm_errno` (resource agent)

`udm_errno` retourne le numéro d'erreur mnoGoSearch, ou bien 0 s'il n'y en a aucune.

Le paramètre agent est un identifiant de résultat, obtenu après un appel à `udm_alloc_agent` .

`udm_errno` retourne le numéro de l'erreur généré par l'agent agent .

8.75.19 `udm_error()` : Message d'erreur mnoGoSearch

string `udm_error` (resource agent)

`udm_errno` retourne le message d'erreur mnoGoSearch, ou une chaîne vide s'il n'y en a pas.

Le paramètre agent est un identifiant de résultat, obtenu après un appel à `udm_alloc_agent` .

`udm_error` retourne le numéro de l'erreur généré par l'agent agent .

8.75.20 `udm_find()` : Effectue une recherche

resource `udm_find` (resource agent , string query)

`udm_add_search_limit` retourne une ressource de résultats en cas de succès, et FALSE en cas d'erreur.

La recherche en elle-même. Le premier argument agent est la session, le second est la requête (query). Pour rechercher, entrez les mots avec lesquels vous voulez faire une recherche, puis cliquez sur le bouton d'envoi. Par exemple, " mysql odbc ". Vous ne devez pas utiliser de guillemets doubles " , car ils sont utilisés par mnoGoSearch pour séparer une requête en mots. Avec l'exemple ci-dessus, mnoGoSearch va rechercher les pages contenant "mysql" et/ou "odbc". Les meilleures réponses seront classées en premier, et affichées en tête de liste. Si vous sélectionnez le mode de recherche "tous" ("ALL"), la recherche va retourner les documents qui contiennent l'un ou l'autre des mots que vous avez entré. Dans le cas où vous utilisez le mode "ANY", la recherche retourne la liste des documents qui contiennent l'un ou l'autre des mots. Si vous voulez accéder aux fonctions avancées de recherche, vous pouvez utiliser le mode "BOOL", qui vous permet d'entrer directement des requêtes.

mnoGoSearch utilise les opérateurs booléens suivants :

& - AND, ET logique. Par exemple, "mysql & odbc". mnoGoSearch recherche toutes les URL qui contiennent à la fois les mots "mysql" et "odbc".

| - OR, OU logique. Par exemple, "mysql | odbc". mnoGoSearch recherche toutes les URL qui contiennent soit "mysql", soit "odbc".

~ - NOT, NON logique. Par exemple, "mysql & ~odbc". mnoGoSearch recherche toutes les URL qui contiennent le mot "mysql" mais ne contiennent pas le mot "odbc". Attention : la requête "~odbc" ne trouvera rien !

() - Groupement de commandes pour les requêtes complexes : par exemple, "(mysql | msql) & ~postgres". Le mode par requête est simple et puissant à la fois. Vous pouvez utiliser les commandes booléennes habituelles avec ce mode.

8.75.21 `udm_free_agent()` : Détruit une session mnoGoSearch

int `udm_free_agent` (resource agent)

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre agent est l'identifiant agent, obtenu après un appel à `udm_alloc_agent` .

`udm_free_agent` détruit l'agent de recherche créé par `udm_alloc_agent` .

8.75.22 `udm_free_ispell_data()` : Libère la mémoire allouée pour ispell par MnoGoSearch

bool `udm_free_ispell_data` (int agent)

`udm_free_ispell_data` retourne toujours TRUE .

agent - Agent mnoGoSearch obtenu après un appel à `udm_alloc_agent` .

Note

<code>udm_free_ispell_data</code> est supportée à partir de la version 3.1.12 de mnoGoSearch et elle ne fait strictement rien avec les versions précédentes.
--

8.75.23 `udm_free_res()` : Libère un résultat mnoGoSearch

bool `udm_free_res` (resource res)

`udm_free_res` libère la mémoire de tous les résultats générés.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre res est un identifiant de résultat, obtenu après un appel à `udm_find` .

8.75.24 `udm_get_doc_count()` : Lit le nombre total de documents dans les bases MnoGoSearch

int `udm_get_doc_count` (resource agent)

udm_get_doc_count retourne le nombre de documents dans les bases de données.

agent - Agent mnoGoSearch obtenu après un appel à udm_alloc_agent .

Note

<u>udm_get_doc_count</u> est supportée à partir de la version mnoGoSearch 3.1.11 ou plus récent.
--

8.75.25 udm_get_res_field() : Lit un champ de résultat mnoGoSearch

string **udm_get_res_field** (resource res , int row , int field)

udm_alloc_agent retourne la valeur du champ field dans la ligne row , du résultat res , et FALSE sinon.

Le paramètre res est un identifiant de résultat, obtenu après un appel à udm_find .

Le paramètre row est le numéro du lien dans la page courante. Il peut valoir de 0 jusqu'à UDM_PARAM_NUM_ROWS-1 .

Le paramètre field est l'identifiant de champ, et peut prendre l'une des valeurs suivantes :

- UDM_FIELD_URL : Champ URL
- UDM_FIELD_CONTENT : Champ "Content-type" (par exemple, "text/html").
- UDM_FIELD_CATEGORY - Champ catégorie du document. Utilisez udm_cat_path pour récupérer le chemin complet de la catégorie courante depuis le dossier racine. (Ce paramètre est valable uniquement depuis la version 4.0.6 de PHP et suivantes.)
- UDM_FIELD_TITLE : Titre du document.
- UDM_FIELD_KEYWORDS : Mots-clés du document (balise META KEYWORDS).
- UDM_FIELD_DESC : Description du document (balise META DESCRIPTION).
- UDM_FIELD_TEXT : Corps du document (balise body, les premières lignes pour donner une idée du document).
- UDM_FIELD_SIZE : Taille du document.
- UDM_FIELD_URLID : Identifiant unique de l'URL.
- UDM_FIELD_RATING : Score de la page (calculé par mnoGoSearch).
- UDM_FIELD_MODIFIED : Date de modification au format unixtimestamp.
- UDM_FIELD_ORDER : Le nombre de documents trouvés.
- UDM_FIELD_CRC : La valeur CRC du document.

8.75.26 udm_get_res_param() : Lit les paramètres de résultats mnoGoSearch

string **udm_get_res_param** (resource res , int param)

udm_get_res_param retourne les paramètres de résultats en cas de succès, FALSE en cas d'erreur.

Le paramètre res est un identifiant de résultat, obtenu après un appel à udm_find .

Le paramètre param peut prendre les valeurs suivantes :

- UDM_PARAM_NUM_ROWS - Nombre de liens trouvés dans le groupe de résultat courant. C'est la valeur de UDM_PARAM_PAGE_SIZE pour tous les groupes, sauf le dernier.

- UDM_PARAM_FOUND - Nombre total de résultats trouvés.
- UDM_PARAM_WORDINFO - Informations sur les mots trouvés, c'est-à-dire que la recherche "un bon livre" retournera "un: stopword, bon:5637, livre: 120"
- UDM_PARAM_SEARCHTIME - Temps de recherche en secondes
- UDM_PARAM_FIRST_DOC - Le numéro du premier document affiché dans le groupe.
- UDM_PARAM_LAST_DOC - Le numéro du dernier document affiché dans le groupe.

8.75.27 `udm_hash32()` : Retourne une somme de contrôle Hash32

int `udm_hash32` (resource agent , string str)

`udm_hash32` prend la chaîne str et retourne un nombre presque unique. Cette fonction requiert un agent mnoGoSearch dans le paramètre agent .

Voir aussi `udm_alloc_agent` .

8.75.28 `udm_load_ispell_data()` : Charge les données ispell

bool `udm_load_ispell_data` (resource agent , int var , string val1 , string val2 , int flag)

`udm_load_ispell_data` charge des données ispell. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

agent - Agent mnoGoSearch obtenu après un appel à `udm_alloc_agent` .

var - Paramètre indiquant la source des données ispell.

Après avoir utilisé cette fonction, pensez à libérer les données de la mémoire avec `udm_free_ispell_data` , même si vous utilisez le mode UDM_ISPELL_TYPE_SERVER .

Le mode le plus rapide est UDM_ISPELL_TYPE_SERVER . UDM_ISPELL_TYPE_TEXT est plus lent, et UDM_ISPELL_TYPE_DB est le plus lent. Ce classement est vrai pour mnoGoSearch 3.1.10 - 3.1.11. Il est prévu d'accélérer le mode DB dans les versions futures, et cela sera plus rapide que le mode TEXT.

- UDM_ISPELL_TYPE_DB indique que les données ispell doivent être chargées depuis la base SQL. Dans ce cas, les paramètres val1 et val2 sont ignorés et doivent être laissés vides. flag doit valoir 1 .

Note

flag indique qu'après le chargement des données ispell à partir de la source, elles doivent être triées (c'est nécessaire au bon fonctionnement d'ispell). Dans le cas où vous chargez les données depuis un fichier, il peut y avoir plusieurs appels à `udm_load_ispell_data` , et il ne vaut pas la peine de trier les valeurs après chaque appel, mais uniquement à la fin. Etant donné qu'en mode DB, toutes les données sont chargées en une seule fois, ce paramètre doit avoir la valeur de 1 . Dans ce mode, en cas d'erreur, par exemple si la table ispell est absente, la fonction retournera FALSE et le code d'erreur, avec son message, seront accessibles avec `udm_error` et `udm_errno` .

Exemple avec `udm_load_ispell_data`

```
<?php
if (! udm_load_ispell_data($udm,UDM_ISPELL_TYPE_DB,'',' ',1)) {
    printf("Erreur #d: '%s'\n", udm_errno($udm), udm_error($udm));
}
```

```

    exit;
}
?>

```

- UDM_ISPELL_TYPE_AFFIX indique que les données ispell doivent être chargées depuis un fichier et initie le chargement. Dans ce cas, val1 définit le code de langue en deux lettres, et val2 est le chemin jusqu'aux fichiers. Notez que si vous utilisez un chemin relatif, le module recherche les fichiers non pas dans UDM_CONF_DIR , mais directement avec le chemin courant, où le script est exécuté. En cas d'erreur avec ce mode, si le fichier est absent, la fonction retourne FALSE , et un message d'erreur sera affiché. Les messages d'erreur ne sont pas accessibles avec `udm_error` et `udm_errno` , puisque ces fonctions ne traitent que les messages SQL. Reportez-vous à la description du paramètre flag .

Exemple avec `udm_load_ispell_data` et UDM_ISPELL_TYPE_AFFIX

```

<?php
if ((! udm_load_ispell_data($udm,UDM_ISPELL_TYPE_AFFIX,'en','/opt/ispell/en.aff',0) ||
    (! udm_load_ispell_data($udm,UDM_ISPELL_TYPE_AFFIX,'ru','/opt/ispell/ru.aff',0) ||
    (! udm_load_ispell_data($udm,UDM_ISPELL_TYPE_SPELL,'en','/opt/ispell/en.dict',0) ||
    (! udm_load_ispell_data($udm,UDM_ISPELL_TYPE_SPELL,'ru','/opt/ispell/ru.dict',1))) {
    exit;
}
?>

```

Note

flag prend la valeur 1 si c'est le dernier appel à cette fonction.

- UDM_ISPELL_TYPE_SPELL indique que les données ispell doivent être chargées depuis un fichier, et initie le chargement du dictionnaire. Dans ce cas, val1 définit le code langue sur deux lettres, et val2 le chemin du fichier. Notez que si vous utilisez un chemin relatif, le module recherche les fichiers non pas dans UDM_CONF_DIR , mais directement avec le chemin courant, où le script est exécuté. En cas d'erreur avec ce mode, si le fichier est absent, la fonction retourne FALSE , et un message d'erreur sera affiché. Les messages d'erreur ne sont pas accessibles avec `udm_error` et `udm_errno` , puisque ces fonctions ne traitent que les messages SQL. Reportez-vous à la description du paramètre flag dans UDM_ISPELL_TYPE_DB.

Exemple avec `udm_load_ispell_data` et UDM_ISPELL_TYPE_SPELL

```

<?php
if ((! Udm_Load_Ispell_Data($udm,UDM_ISPELL_TYPE_AFFIX,'en','/opt/ispell/en.aff',0) ||
    (! Udm_Load_Ispell_Data($udm,UDM_ISPELL_TYPE_AFFIX,'ru','/opt/ispell/ru.aff',0) ||
    (! Udm_Load_Ispell_Data($udm,UDM_ISPELL_TYPE_SPELL,'en','/opt/ispell/en.dict',0) ||
    (! Udm_Load_Ispell_Data($udm,UDM_ISPELL_TYPE_SPELL,'ru','/opt/ispell/ru.dict',1) ) ) {
    exit;
}
?>

```

Note

flag prend la valeur 1 si c'est le dernier appel à cette fonction.

- UDM_ISPELL_TYPE_SERVER active le support des serveurs ispell. val1 indique alors l'adresse de l'hôte qui supporte le serveur ispell. val2 n'est pas encore utilisé, mais dans les cas futurs, il indiquera le numéro de port utilisé par le serveur ispell. flag n'est pas utile, car les données sont déjà triées.

Les serveurs Speld lisent les données d'orthographe dans une configuration séparée (par défaut `/usr/local/mnogosearch/etc/speld.conf`), les trie et les stocke en mémoire. Avec les clients, le serveur communique de deux façons : vers les indexeurs, tout le contenu de la mémoire est transféré pour que l'indexeur travaille plus vite ; vers le moteur de recherche, il reçoit les mots à normaliser et les rend au client corrigés. Cela permet une plus grande rapidité d'exécution, en comparaison des modes db et text (notamment, les tris et les

chargements sont beaucoup plus rapides).

`udm_load_ispell_data` en mode `UDM_ISPELL_TYPE_SERVER` ne charge pas vraiment les données ispell, mais définit simplement l'adresse du serveur. En fait, le serveur sera automatiquement utilisé par `udm_find` lors des recherches. En cas d'erreur, (par exemple si le serveur ispell ne fonctionne pas ou que l'hôte indiqué est invalide), la conversion sera annulée, mais aucun message d'erreur ne sera affiché.

Note
Cette fonction est disponible à partir de mnoGoSearch 3.1.12.
Exemple avec <code>udm_load_ispell_data</code> et <code>UDM_ISPELL_TYPE_SERVER</code>
<pre><?php if (!udm_load_ispell_data(\$udm,UDM_ISPELL_TYPE_SERVER,','',1)) { echo "Erreur au chargement de la bibliothèque ispell sur le serveur
\n"; exit; } ?></pre>

8.75.29 `udm_open_stored()` : Ouvre une connexion MnoGoSearch avec un document stocké

`int udm_open_stored (resource agent , string storedaddr)`

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.75.30 `udm_set_agent_param()` : Modifie les paramètres de l'agent MnoGoSearch

`bool udm_set_agent_param (resource agent , int var , string val)`

`udm_set_agent_param` définit les paramètres de l'agent mnoGoSearch. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Les paramètres suivants et leurs valeurs sont disponibles :

- `UDM_PARAM_PAGE_NUM` - Utilisé pour choisir le numéro de groupe de résultat (les résultats sont retournés par groupe, commençant à 0, avec `UDM_PARAM_PAGE_SIZE` résultats par page).
- `UDM_PARAM_PAGE_SIZE` - Nombre de résultats affichés par page.
- `UDM_PARAM_SEARCH_MODE` - Mode de recherche. Les valeurs suivantes sont disponibles : `UDM_MODE_ALL` - recherche tous les mots ; `UDM_MODE_ANY` - recherche l'un des mots ; `UDM_MODE_PHRASE` - recherche une phrase ; `UDM_MODE_BOOL` - recherche booléenne. Voir `udm_find` pour plus de détails sur les recherches booléennes.
- `UDM_PARAM_CACHE_MODE` - Active/désactive le cache. Lorsque le cache est activé, le moteur de recherche va stocker les résultats sur le disque. Lorsque deux requêtes seront similaires, il pourra retourner les résultats plus rapidement, sans recherche. Valeurs disponibles : `UDM_CACHE_ENABLED` , `UDM_CACHE_DISABLED` .
- `UDM_PARAM_TRACK_MODE` - Active le mode de suivi de requête. Depuis la version 3.1.2, mnoGoSearch dispose d'un suivi de requête. Notez que ce suivi n'est implémenté qu'avec les versions SQL et n'est pas disponible avec les bases de données intégrées. Pour

utiliser ce suivi, vous devez créer des tables de suivi. Pour mysql, utilisez le script create/mysql/track.txt . Lorsque vous effectuez une recherche avec l'interface, ces tables stockeront les mots recherchés ainsi que le nombre de mots trouvés, et la date. Valeurs disponibles : UDM_TRACK_ENABLED , UDM_TRACK_DISABLED .

- UDM_PARAM_PHRASE_MODE - indique si les index des bases de données utilisent des phrases (paramètre "phrase" dans indexer.conf). Valeurs disponibles : UDM_PHRASE_ENABLED et UDM_PHRASE_DISABLED . Notez bien que si la recherche par phrase est activée (UDM_PHRASE_ENABLED), il est toujours possible de faire des recherches dans d'autres modes, (ANY, ALL, BOOL ou PHRASE). En version 3.1.10 de mnoGoSearch, la recherche par phrase n'est supportée que pour les modes SQL et intégré, tandis qu'en 3.1.11, la recherche par phrase est supportée par le mode cache.

Exemple de recherche par phrase :

"Arizona desert" - Cette requête retourne tous les documents qui contiennent les mots "Arizona desert" comme une phrase. Notez que vous devez mettre des guillemets doubles autour des phrases.

- UDM_PARAM_CHARSET - Définit le jeu de caractères local. Valeurs disponibles : tous les jeux supportés par mnoGoSearch. koi8-r, cp1251, ...
- UDM_PARAM_STOPFILE - Définit le nom et le chemin du fichier de mots ignorés. Il y a une petite différence avec mnoGoSearch : avec mnoGoSearch, si le chemin est NULL ou relatif, il est utilisé à partir de UDM_CONF_DIR , alors qu'en PHP, le module va rechercher à partir du chemin courant, c'est-à-dire celui du script courant.
- UDM_PARAM_STOPTABLE - Charge la liste des mots ignorés depuis une table SQL. Vous pouvez utiliser plusieurs tables SQL. Cette commande n'a aucun effet si mnoGoSearch n'a pas été compilé avec le support de base de données.
- UDM_PARAM_WEIGHT_FACTOR - Représente le poids relatif des différentes parties d'un document. Actuellement, le corps, titre, mots-clés, descriptions et url sont supportés. Pour activer cette fonctionnalité, utilisez le degré 2 de *Weight commands, dans le fichier indexer.conf . Imaginons que vous avez choisi les poids suivants :

```
URLWeight      1
BodyWeight     2
TitleWeight    4
KeywordWeight  8
DescWeight     16
```

Comme l'indexeur utilise l'opérateur de bits OR pour mesurer le poids des mots, il est possible que le même mot soit trouvé plusieurs fois dans le même document lors des recherches. Un mot qui n'apparaît qu'une fois dans le corps sera défini par 00000010 (notation binaire). Un mot qui apparaîtra dans plusieurs parties pourra avoir la notation 00011111.

La valeur de ce paramètre est une chaîne de chiffres hexadécimaux, sous la forme ABCDE . Chaque chiffre est un facteur correspondant à un poids affecté à une partie du document. Por la situation décrite ci-dessus,

```
E est le facteur de poids 1  (URL Weight bit)
D est le facteur de poids 2  (BodyWeight bit)
C est le facteur de poids 4  (TitleWeight bit)
B est le facteur de poids 8  (KeywordWeight bit)
A est le facteur de poids 16 (DescWeight bit)
```

Exemples :

UDM_PARAM_WEIGHT_FACTOR =00001 ne recherche que dans les URL.

UDM_PARAM_WEIGHT_FACTOR =00100 ne recherche que dans les Titres.

UDM_PARAM_WEIGHT_FACTOR =11100 recherche dans les Titres, Mots-clés, Description mais pas dans le Corps ou les URL.

UDM_PARAM_WEIGHT_FACTOR =F9421 recherche dans :

```
Description avec un poids de 15 (F hex)
Mots-clés avec un poids de 9
Titre avec un poids de 4
Corps avec un poids de 2
URL avec un poids de 1
```

Si UDM_PARAM_WEIGHT_FACTOR est omis, la valeur par défaut est utilisée.

- UDM_PARAM_WORD_MATCH - Recherche des mots. Vous pouvez utiliser ce paramètre pour choisir le type de recherche de mots. Cette fonctionnalité n'est valable qu'en mode "single" et "multi", avec les bases SQL ou intégrée. Elle ne fonctionne pas en mode intégré, ni avec d'autres modes, car les CRC ne supportent pas les recherches de sous-chaînes. Les valeurs disponibles sont :

UDM_MATCH_BEGIN - début de mot;

UDM_MATCH_END - fin de mot;

UDM_MATCH_WORD - tout le mot;

UDM_MATCH_SUBSTR - une sous-partie de mot.

- UDM_PARAM_MIN_WORD_LEN - définit les tailles extrêmes de mots. Tout mot plus court que la limite inférieure est ignoré. Notez que ce paramètre est inclusif, c'est-à-dire que si UDM_PARAM_MIN_WORD_LEN =3, un mot de 3 caractères ne sera pas ignoré, alors qu'un mot de 2 caractères le sera. Par défaut, la valeur est de 1.
- UDM_PARAM_ISPELL_PREFIXES - Valeurs possibles : UDM_PREFIXES_ENABLED et UDM_PREFIXES_DISABLED . Ces valeurs activent et désactivent le support des préfixes. Par exemple, si le mot "testé" est placé dans la requête de recherche, les mots tels "test", "tester", etc. seront aussi recherchés. Les suffixes sont supportés par défaut. Les préfixes modifient généralement le sens des mots. Par exemple, si vous cherchez "testé", vous ne souhaitez pas trouver "protesté" ou "contesté". Le support des préfixes peut cependant être utilisé pour des raisons d'orthographe. Pour activer ispell, vous devez charger les données ispell avec la fonction `udm_load_ispell_data` .
- UDM_PARAM_CROSS_WORDS - Active ou désactive le support "CROSS_WORDS". Valeurs possibles : UDM_CROSS_WORDS_ENABLED et UDM_CROSS_WORDS_DISABLED .

La fonctionnalité "CROSS_WORDS" vous permet d'effectuer des recherches dans les balises (entre `` ``), pour utiliser le nom du lien. Ce mode fonctionne avec les bases de données SQL et n'est pas supporté par les modes intégrés ou le cache.

Note

CROSS_WORDS est supporté à partir de mnoGoSearch 3.1.11.
--

- UDM_PARAM_VARDIR - spécifie un chemin précis sur le disque où l'indexeur enregistre les données lorsqu'il utilise le cache et les bases de données internes. Par défaut, le dossier /var de l'installation de **MNOGOSEARCH** est utilisé. Ce paramètre est disponible en PHP

4.1.0 et plus récent.

8.76 Sessions Mohawk

8.76.1 Introduction

msession est une interface avec un démon de sessions rapides, qui permet de gérer les sessions localement ou à distance. Il est conçu pour fournir une gestion de sessions cohérente pour une ferme de serveurs PHP. Plus d'informations sur les msession et le démon sont disponibles sur le site <http://devel.mohawksoft.com/msession.html> .

Note

Cette extension n'est pas disponible sur les plates-formes Windows.

8.76.2 Pré-requis

8.76.3 Installation

Pour activer le support de l'extension msession, vous devez compiler PHP avec l'option --with-msession[=DIR] , où DIR est le dossier d'installation de msession.

8.76.4 Configuration à l'exécution

8.76.5 Types de ressources

8.76.6 Constantes pré-définies

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [msession_connect](#)
- [msession_count](#)
- [msession_create](#)
- [msession_destroy](#)
- [msession_disconnect](#)
- [msession_find](#)
- [msession_get_array](#)
- [msession_get_data](#)
- [msession_get](#)
- [msession_inc](#)
- [msession_list](#)
- [msession_listvar](#)
- [msession_lock](#)
- [msession_plugin](#)
- [msession_randstr](#)
- [msession_set_array](#)

- [msession_set_data](#)
- [msession_set](#)
- [msession_timeout](#)
- [msession_uniq](#)
- [msession_unlock](#)

8.76.8 msession_count() : Compte le nombre de sessions

int **msession_count** (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.9 msession_create() : Crée une session

bool **msession_create** (string session)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.10 msession_destroy() : Détruit la session

bool **msession_destroy** (string name)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.11 msession_disconnect() : Ferme la connexion au serveur msession

void **msession_disconnect** (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.12 msession_find() : Trouve toutes les sessions avec un nom et une valeur

array **msession_find** (string name , string value)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.13 msession_get_array() : Lit un tableau de variables msession

array **msession_get_array** (string session)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.14 `msession_get_data()` : Lit les données de la session

string `msession_get_data` (string session)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.15 `msession_get()` : Lit une valeur dans une session

string `msession_get` (string session , string name , string value)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.16 `msession_inc()` : Incrmente une valeur dans une session

string `msession_inc` (string session , string name)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.17 `msession_list()` : Liste toutes les sessions

array `msession_list` (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.18 `msession_listvar()` : Liste les sessions contenant une variable

array `msession_listvar` (string name)

`msession_listvar` retourne un tableau associatif des valeurs que prend la variable de session name dans les différentes sessions en cours.

Utilisée pour rechercher les sessions ayant des attributs communs.

8.76.19 `msession_lock()` : Verrouille une session

int `msession_lock` (string name)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.20 `msession_plugin()` : Appelle une fonction d'échappement dans les plugins de personnalité msession

string `msession_plugin` (string session , string val , string param)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.21 `msession_randstr()` : Lit une chaîne aléatoire

string `msession_randstr` (int param)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.22 `msession_set_array()` : Définit les variables `msession` depuis un tableau

void `msession_set_array` (string session , array tuples)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.23 `msession_set_data()` : Définit les données de session

bool `msession_set_data` (string session , string value)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.24 `msession_set()` : Modifie une valeur dans la session

bool `msession_set` (string session , string name , string value)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.25 `msession_timeout()` : Lit et/ou modifie la durée de vie de la session

int `msession_timeout` (string session , int param)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.26 `msession_uniq()` : Récupère un identifiant unique

string `msession_uniq` (int param)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.76.27 `msession_unlock()` : Déverrouille une session

int `msession_unlock` (string `session` , int `key`)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.77 mSQL

8.77.1 Introduction

Ces fonctions vous permettent d'accéder aux bases de données mSQL. Plus d'informations sur mSQL à <http://www.hughes.com.au/>.

8.77.2 Installation

Pour cela, vous devez compiler PHP avec le support msql, en utilisant l'option de configuration `--with-msql[=DIR]`. Par défaut, le chemin DIR est `/usr/local/msql3`.

Note
Note aux utilisateurs Win32
Afin d'activer ce module dans l'environnement Windows, vous devez copier les bibliothèques <code>msql.dll</code> depuis le dossier DLL de PHP/Win32 dans le dossier système SYSTEM32 de votre machine (par exemple : <code>C:\WINNT\SYSTEM32</code> ou <code>C:\WINDOWS\SYSTEM32</code>).

8.77.3 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable	Historique
<code>msql.allow_persistent</code>	"1"	PHP_INI_ALL	
<code>msql.max_persistent</code>	"-1"	PHP_INI_ALL	
<code>msql.max_links</code>	"-1"	PHP_INI_ALL	

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

`msql.allow_persistent` bool

Activation ou non des connexions mSQL persistantes.

`msql.max_persistent` entier

Le nombre maximum de connexions mSQL persistantes par processus.

`msql.max_links` entier

Le nombre maximum de connexions mSQL par processus, y compris les connexions persistantes.

8.77.4 Types de ressources

Il y a deux types de ressources utilisées dans le module mSQL. Le premier est le lien identifiant la connexion de la base de données, le second, représente le résultat d'une requête.

8.77.5 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

MSQL_ASSOC ([entier](#))
 MSQL_NUM ([entier](#))
 MSQL_BOTH ([entier](#))

8.77.6 Exemples

Cet exemple simple vous présente comment vous connecter à une base mSQL pour y lire quelques lignes.

Exemple d'utilisation avec mSQL

```
<?php
/* Connexion à la base, sélection de la base de données */
$link = msql_connect('localhost', 'username', 'password')
  or die('Could not connect : ' . msql_error($link));

msql_select_db('database', $link)
  or die('Could not select database');

/* Soumission de la requête */
$query = 'SELECT * FROM my_table';
$result = msql_query($query, $link) or die('Query failed : ' . msql_error($link));

/* Affichage des résultats */
echo "<table>\n";
while ($row = msql_fetch_array($result, MSQL_ASSOC)) {
  echo "\t<tr>\n";
  foreach ($row as $col_value) {
    echo "\t\t<td>$col_value</td>\n";
  }
  echo "\t</tr>\n";
}
echo "</table>\n";

/* Libération des résultats */
msql_free_result($result);

/* Déconnexion */
msql_close($link);
?>
```

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [msql_affected_rows](#)
- [msql_close](#)
- [msql_connect](#)
- [msql_create_db](#)
- [msql_createdb](#)
- [msql_data_seek](#)
- [msql_db_query](#)
- [msql_dbname](#)

- [mysql_drop_db](#)
- [mysql_error](#)
- [mysql_fetch_array](#)
- [mysql_fetch_field](#)
- [mysql_fetch_object](#)
- [mysql_fetch_row](#)
- [mysql_field_flags](#)
- [mysql_field_len](#)
- [mysql_field_name](#)
- [mysql_field_seek](#)
- [mysql_field_table](#)
- [mysql_field_type](#)
- [mysql_fieldflags](#)
- [mysql_fieldlen](#)
- [mysql_fieldname](#)
- [mysql_fieldtable](#)
- [mysql_fieldtype](#)
- [mysql_free_result](#)
- [mysql_list_dbs](#)
- [mysql_list_fields](#)
- [mysql_list_tables](#)
- [mysql_num_fields](#)
- [mysql_num_rows](#)
- [mysql_numfields](#)
- [mysql_numrows](#)
- [mysql_pconnect](#)
- [mysql_query](#)
- [mysql_regcase](#)
- [mysql_result](#)
- [mysql_select_db](#)
- [mysql_tablename](#)
- [mysql](#)

8.77.8 [mysql_close\(\)](#) : Ferme une connexion mSQL

bool **mysql_close** (*resource link_identifier*)

[mysql_close](#) ferme la connexion non persistante au serveur de base de données mSQL associée au paramètre `link_identifier` .

Notez bien qu'il n'est pas toujours nécessaire d'appeler cette fonction, car les connexions non persistantes seront automatiquement fermées à la fin du script. Voir aussi la [libération des ressources](#) .

8.77.9 [mysql_connect\(\)](#) : Ouvre une connexion mSQL

resource **mysql_connect** (*string hostname*)

[mysql_connect](#) établit une connexion à un serveur mSQL.

Si un deuxième appel est fait à [mysql_connect](#) , avec les mêmes arguments, ce ne sera pas une nouvelle connexion qui va être ouverte, mais l'ancienne connexion qui sera utilisée, et son

identifiant sera retourné.

Le lien au serveur sera fermé dès la fin du script, ou bien, manuellement, lors de l'appel de mysql_close .

8.77.10 mysql_create_db() : Crée une base de données mSQL

bool **mysql_create_db** (string database_name , resource link_identifieur)

mysql_create_db essaie de créer une nouvelle base de données nommée database_name sur le serveur référencé par l'identifiant link_identifieur .

8.77.11 mysql_createdb() : Alias de mysql_create_db

Cette fonction est un alias de : mysql_create_db .

8.77.12 mysql_data_seek() : Déplace le pointeur interne mSQL

bool **mysql_data_seek** (resource result , int row_number)

mysql_data_seek déplace le pointeur interne de résultat mSQL, et le place à l'offset donné. Le prochain appel à la fonction mysql_fetch_row retournera cette ligne.

8.77.13 mysql_db_query() : Soumet une requête mSQL

resource **mysql_db_query** (string database , string query , resource link_identifieur)

mysql_db_query sélectionne une base de données, et exécute une requête.

8.77.14 mysql_dbname() : Alias de mysql_result

Cette fonction est un alias de : mysql_result .

8.77.15 mysql_drop_db() : Efface une base de données mSQL

bool **mysql_drop_db** (string database_name , resource link_identifieur)

mysql_drop_db essaie d'effacer une base de données entière sur le serveur référencé par l'identifiant link_identifieur .

8.77.16 mysql_error() : Retourne le message d'erreur mSQL

string **mysql_error** (void)

mysql_error retourne la dernière erreur retournée par le serveur mSQL. Notez que seul le dernier message est retourné par la fonction mysql_error .

8.77.17 `mysql_fetch_array()` : Lit une ligne sous la forme d'un tableau

array `mysql_fetch_array` (resource result , int result_type)

`mysql_fetch_array` est une version évoluée de `mysql_fetch_row` . En plus d'enregistrer les données dans un tableau à indice numérique, il enregistre aussi les données dans un tableau associatif, en utilisant les noms des champs comme clés.

Il est important de noter que `mysql_fetch_array` n'est PAS significativement plus lent que `mysql_fetch_row` , alors qu'elle apporte un confort d'utilisation appréciable.

8.77.18 `mysql_fetch_field()` : Lit la valeur d'un champ mSQL

object `mysql_fetch_field` (resource result , int field_offset)

`mysql_fetch_field` sert à lire les informations sur les champs dans certaines requêtes.

8.77.19 `mysql_fetch_object()` : Lit une ligne sous la forme d'un objet

object `mysql_fetch_object` (resource result)

`mysql_fetch_object` est identique à `mysql_fetch_array` , avec une différence : c'est un objet qui est retourné, à la place d'un tableau. Par conséquent, cela signifie que vous ne pouvez accéder aux valeurs que par les noms des champs, et non plus avec leur offset. (Les nombres sont interdits dans les noms de propriétés).

`mysql_fetch_object` est aussi rapide que `mysql_fetch_array` , et marginalement plus lente que `mysql_fetch_row` (la différence n'est pas significative).

8.77.20 `mysql_fetch_row()` : Retourne une ligne sous la forme d'un tableau

array `mysql_fetch_row` (resource result)

`mysql_fetch_row` récupère une ligne de données depuis le résultat associé avec l'identifiant de requête spécifié. La ligne est retournée sous la forme d'un tableau. Chaque colonne du résultat est stockée dans une ligne du tableau, en commençant à la position 0.

Les appels suivants à la fonction `mysql_fetch_row` retourneront les prochaines lignes du jeu de résultat, ou FALSE s'il n'y a plus de lignes de disponible.

8.77.21 `mysql_field_flags()` : Lit les options de champ mysql

string `mysql_field_flags` (resource result , int field_offset)

`mysql_field_flags` retourne les options du champ `field_offset` .

8.77.22 mysql_field_len() : Lit la taille du champ mSQL

int **mysql_field_len** (resource result , int field_offset)

mysql_field_len retourne la taille du champ spécifié.

8.77.23 mysql_field_name() : Lit le nom du champ mysql

string **mysql_field_name** (resource result , int field_offset)

mysql_field_name retourne le nom du champ de l'index spécifié.

8.77.24 mysql_field_seek() : Fixe l'offset d'un champ mSQL

bool **mysql_field_seek** (resource result , int field_offset)

mysql_field_seek recherche l'offset du champ field_offset . Le prochain appel à mysql_fetch_field sans l'argument field_offset , retournera ce champ.

8.77.25 mysql_field_table() : Lit le nom de la table du champ mysql

int **mysql_field_table** (resource result , int field_offset)

mysql_field_table retourne le nom de la table dans laquelle le champ field a été lu.

8.77.26 mysql_field_type() : Lit le type de champ mSQL

string **mysql_field_type** (resource result , int field_offset)

mysql_field_type récupère le type du champ spécifié par son index.

8.77.27 mysql_fieldflags() : Alias de mysql_field_flags

Cette fonction est un alias de la fonction mysql_field_flags .

8.77.28 mysql_fieldlen() : Alias de mysql_field_len

Cette fonction est un alis de la fonction mysql_field_len .

8.77.29 mysql_fieldname() : Alias de mysql_field_name

Cette fonction est un alias de la fonction mysql_field_name .

8.77.30 `mysql_fieldtable()` : Alias de `mysql_field_table`

Cette fonction est un alias de la fonction `mysql_field_table` .

8.77.31 `mysql_fieldtype()` : Alias de `mysql_field_type`

Cette fonction est un alias de la fonction `mysql_field_type` .

8.77.32 `mysql_free_result()` : Libère le résultat de la mémoire

`bool mysql_free_result (resource result)`

`mysql_free_result` libère de la mémoire le résultat associé à l'identifiant de résultat `query_identifier` . Lorsque PHP a terminé une requête, cette mémoire est libérée, ce qui fait que vous n'aurez pas besoin de cette fonction. Vous pouvez toujours l'utiliser pour vous assurer que vous n'utilisez pas trop de mémoire durant un script.

8.77.33 `mysql_list_dbs()` : Liste les bases de données MySQL sur un serveur

`resource mysql_list_dbs (resource link_identifier)`

`mysql_list_dbs` liste tous les bases de données disponibles sur la base spécifiée par `link_identifier` .

8.77.34 `mysql_list_fields()` : Liste les champs d'une table

`resource mysql_list_fields (string database , string tablename , resource link_identifier)`

`mysql_list_fields` lit les informations de la table `tablename` .

8.77.35 `mysql_list_tables()` : Liste les tables MySQL sur une base de données

`resource mysql_list_tables (string database , resource link_identifier)`

`mysql_list_tables` liste les tables de la base de données spécifiée par `database` .

8.77.36 `mysql_num_fields()` : Retourne le nombre de champs dans un résultat

`int mysql_num_fields (resource result)`

`mysql_num_fields` retourne le nombre de champs du résultat associé à l'identifiant `query_identifier` .

8.77.37 `mysql_num_rows()` : Retourne le nombre de lignes dans un résultat

int `mysql_num_rows` (resource query_identifieur)

`mysql_num_rows` retourne le nombre de lignes dans d'un jeu de résultat.

8.77.38 `mysql_numfields()` : Alias de `mysql_num_fields`

Cette fonction est un alias de la fonction `mysql_num_fields` .

8.77.39 `mysql_numrows()` : Alias de `mysql_num_rows`

Cette fonction est un alias de la fonction `mysql_num_rows` .

8.77.40 `mysql_pconnect()` : Ouvre une connexion persistante à un serveur mSQL

resource `mysql_pconnect` (string hostname)

`mysql_pconnect` se comporte presque comme `mysql_connect` mais avec deux différences majeures.

D'abord, lors de la connexion, `mysql_pconnect` cherche si une connexion persistante a déjà été ouverte sur le même hôte. Si une telle connexion est trouvée, elle sera utilisée.

Deuxièmement, la connexion au serveur SQL ne sera pas terminée lors de la fin de l'exécution du script. A la place, le lien restera ouvert pour d'autres connexions futures. (`mysql_close` ne fermera pas un lien ouvert par `mysql_pconnect`).

8.77.41 `mysql_query()` : Envoie une requête mSQL

resource `mysql_query` (string query , resource link_identifieur)

`mysql_query` envoie une requête à la base de données active, sur le serveur associé à l'identifiant de connexion link_identifieur .

8.77.42 `mysql_regcase()` : Alias de `sql_regcase`

Cette fonction est un alias de la fonction `sql_regcase` .

8.77.43 `mysql_result()` : Retourne les données de résultat

string `mysql_result` (resource result , int row , mixed field)

`mysql_result` retourne la valeur de la cellule, à la ligne row et l'offset spécifié, field dans le résultat mSQL query_identifieur .

Lorsque vous travaillez sur des résultats de grande taille, il est préférable d'utiliser les fonctions qui récupèrent toute la ligne (voir ci-dessous). Comme ces fonctions retournent plusieurs cellules en même temps, elles sont beaucoup plus rapide que mysql_result .

Alternatives recommandées : mysql_fetch_row , mysql_fetch_array et mysql_fetch_object .

8.77.44 mysql_select_db() : Sélectionne une base de données mSQL

bool **mysql_select_db** (string database_name , resource link_identifier)

mysql_select_db sélectionne la base de données active sur le serveur associé à link_identifier .

Les appels suivant à mysql_query seront effectués sur la base de données active.

8.77.45 mysql_tablename() : Alias de mysql_result

Cette fonction est un alias de la fonction mysql_result .

8.77.46 mysql() : Alias de mysql_db_query

Cette fonction est un alias de la fonction mysql_db_query .

8.78 Microsoft SQL Server

8.78.1 Introduction

Ces fonctions vous permettent d'accéder aux serveurs de données MS SQL.

8.78.2 Pré-requis

Requis pour les plates-formes Windows

Vous avez besoin d'un client MS SQL Client Tools sur votre système, accessible à PHP. Les Client Tools peuvent être installés depuis le CD MS SQL Server CD ou en copiant ntwdblib.dll depuis le dossier \winnt\system32 du serveur MS SQL vers \winnt\system32 sur le serveur PHP. Copier ntwdblib.dll ne fera que fournir le moyen d'accès. La configuration du client requiert l'installation des autres outils.

Pré-requis pour les plates-formes Unix/Linux.

Pour utiliser l'extension MS SQL sur Unix/Linux, vous devez commencer par compiler et installer la bibliothèque FreeTDS. Le code source et les instructions d'installation sont disponibles sur le site de FreeTDS : <http://www.freetds.org/>

Note

Sur Windows, la librairie DBLIB de Microsoft est utilisée. Les fonctions qui retournent les noms de colonnes sont basées sur la fonction dbcolname() de DBLIB. DBLIB a été développée pour MS SQL Server 6.x où la longueur maximale de l'identifiant est 30. Pour cette raison, la longueur maximale des colonnes est 30. Sur les plates-formes où FreeTDS est utilisé (Linux), il n'y pas ce problème.

8.78.3 Installation

L'extension MSSQL est activée par la ligne `extension=php_mssql.dll` dans le fichier `php.ini` .

Pour que ces fonctions fonctionnent, vous devez compiler PHP avec `--with-mssql[=DIR]` , où DIR est le préfixe d'installation FreeTDS. Et le FreeTDS doit être compilé en utilisant `--enable-msdblib` .

8.78.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable	Historique
<code>mssql.allow_persistent</code>	"1"	PHP_INI_SYSTEM	
<code>mssql.max_persistent</code>	"-1"	PHP_INI_SYSTEM	
<code>mssql.max_links</code>	"-1"	PHP_INI_SYSTEM	
<code>mssql.min_error_severity</code>	"10"	PHP_INI_ALL	
<code>mssql.min_message_severity</code>	"10"	PHP_INI_ALL	
<code>mssql.compatability_mode</code>	"0"	PHP_INI_ALL	
<code>mssql.connect_timeout</code>	"5"	PHP_INI_ALL	

mssql.timeout	"60"	PHP_INI_ALL	Disponible depuis PHP 4.1.0.
mssql.textsize	"-1"	PHP_INI_ALL	
mssql.textlimit	"-1"	PHP_INI_ALL	
mssql.batchsize	"0"	PHP_INI_ALL	Disponible depuis PHP 4.0.4.
mssql.datetimeconvert	"1"	PHP_INI_ALL	Disponible depuis PHP 4.2.0.
mssql.secure_connection	"0"	PHP_INI_SYSTEM	Disponible depuis PHP 4.3.0.
mssql.max_procs	"-1"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [Directives du php.ini](#) .

8.78.5 Types de ressources

8.78.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

[MSSQL_ASSOC](#) ([entier](#))
[MSSQL_NUM](#) ([entier](#))
[MSSQL_BOTH](#) ([entier](#))
[SQLTEXT](#) ([entier](#))
[SQLVARCHAR](#) ([entier](#))
[SQLCHAR](#) ([entier](#))
[SQLINT1](#) ([entier](#))
[SQLINT2](#) ([entier](#))
[SQLINT4](#) ([entier](#))
[SQLBIT](#) ([entier](#))
[SQLFLT8](#) ([entier](#))

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [mssql_bind](#)
- [mssql_close](#)
- [mssql_connect](#)
- [mssql_data_seek](#)
- [mssql_execute](#)
- [mssql_fetch_array](#)
- [mssql_fetch_assoc](#)
- [mssql_fetch_batch](#)
- [mssql_fetch_field](#)
- [mssql_fetch_object](#)
- [mssql_fetch_row](#)
- [mssql_field_length](#)
- [mssql_field_name](#)
- [mssql_field_seek](#)
- [mssql_field_type](#)
- [mssql_free_result](#)

- [mssql_free_statement](#)
- [mssql_get_last_message](#)
- [mssql_guid_string](#)
- [mssql_init](#)
- [mssql_min_error_severity](#)
- [mssql_min_message_severity](#)
- [mssql_next_result](#)
- [mssql_num_fields](#)
- [mssql_num_rows](#)
- [mssql_pconnect](#)
- [mssql_query](#)
- [mssql_result](#)
- [mssql_rows_affected](#)
- [mssql_select_db](#)

8.78.8 [mssql_close\(\)](#) : Ferme une connexion MS SQL Server

bool **mssql_close** (*resource link_identifiant*)

[mssql_close](#) ferme la connexion à la base MS SQL Server, qui était associée à l'identifiant `link_identifiant` . Si ce dernier n'est pas précisé, la dernière connexion ouverte sera fermée.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Notez qu'il n'est pas nécessaire de fermer les connexions non persistantes aux bases de données, car elles seront fermées automatiquement à la fin du script.

[mssql_close](#) ne peut pas fermer les liens persistants, générés par [mssql_pconnect](#) .

Voir aussi [mssql_connect](#) et [mssql_pconnect](#) .

8.78.9 [mssql_connect\(\)](#) : Ouvre une connexion à un serveur MS SQL server

resource **mssql_connect** (*string servername* , *string username* , *string password*)

[mssql_connect](#) retourne un identifiant de lien en cas de succès, et FALSE sinon.

[mssql_connect](#) établit une connexion à un serveur MS SQL. Le nom du serveur `servername` doit être valide, comme défini dans les fichiers d'interface.

Si un deuxième appel est fait à [mssql_connect](#) avec les mêmes arguments, un nouveau lien ne sera pas retourné, mais le lien déjà ouvert sera utilisé.

Le lien avec le serveur sera fermé dès la fin du script, ce qui fait qu'on n'est pas obligé de fermer explicitement la connexion à la fin du script avec [mssql_close](#) .

Voir aussi [mssql_pconnect](#) et [mssql_close](#) .

8.78.10 `mssql_data_seek()` : Déplace le pointeur interne de ligne

`bool mssql_data_seek (resource result_identifier , int row_number)`

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

`mssql_data_seek` déplace le pointeur interne de ligne, dans le résultat `result_identifier` , jusqu'à la ligne `row_number` . Le prochain appel à `mssql_fetch_row` retournera cette ligne.

8.78.11 `mssql_execute()` : Exécute une procédure stockée sur un serveur MS SQL

`mixed mssql_execute (resource stmt , bool skip_results)`

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.
Note
Si la procédure stockée retourne un paramètre ou une valeur, ils seront disponibles après l'appel à la fonction <code>mssql_execute</code> , à moins que la procédure stockée ne retourne plusieurs résultats. Dans ce cas, utilisez la fonction <code>mssql_next_result</code> pour passer en revue tous les résultats. Lorsque le dernier résultat aura été traité, les valeurs des paramètres de sortie et les valeurs retournées seront disponibles.

Voir aussi `mssql_bind` , `mssql_free_statement` et `mssql_init` .

8.78.12 `mssql_fetch_array()` : Lit une ligne de résultat MS SQL dans un tableau

`array mssql_fetch_array (resource result , int result_type)`

`mssql_fetch_array` retourne un tableau qui contient les valeurs de la ligne courante dans le résultat `result` , en cas de succès, et FALSE en cas d'échec.

`mssql_fetch_array` est une version améliorée de `mssql_fetch_row` . En plus de stocker les données dans un tableau à index numérique, elle les stocke aussi dans un tableau associatif, en utilisant les noms de colonnes comme clés.

Une chose importante à noter est que `mssql_fetch_array` n'est PAS significativement plus lente que `mssql_fetch_row` , tandis qu'elle apporte un confort appréciable.

Note
Les noms des champs retournés par cette fonction sont sensibles à la casse .
Note
Cette fonction définit les champs NULL à la valeur PHP NULL .

Pour plus de détails, voyez `mssql_fetch_row` .

8.78.13 `mssql_fetch_assoc()` : Retourne un tableau associatif pour la ligne courant de résultat MS SQL Server

array `mssql_fetch_assoc` (resource result_id)

Attention
 Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.78.14 `mssql_fetch_batch()` : Retourne le prochain lot de lignes MS SQL Server

int `mssql_fetch_batch` (resource result_index)

Attention
 Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.78.15 `mssql_fetch_field()` : Lit les informations sur le champ

object `mssql_fetch_field` (resource result , int field_offset)

`mssql_fetch_field` retourne un objet contenant les informations sur un champ.

`mssql_fetch_field` sert à lire des informations spécifiques à un champ, dans un résultat de requête. Si l'offset du champ `field_offset` n'est pas précisé, le prochain champ sera analysé.

Les propriétés de l'objet sont :

- name - nom de la colonne. Si la colonne est le résultat d'une fonction, le nom de la colonne sera computed#N, où #N est un numéro de série.
- column_source - le nom de la table d'où la colonne est originaire.
- max_length - taille maximale de la colonne
- numeric - 1 si la colonne est numérique
- type - le type de la colonne.

Voir aussi `mssql_field_seek` .

8.78.16 `mssql_fetch_object()` : Retourne une ligne de résultat MS SQL Server sous la forme d'un objet

object `mssql_fetch_object` (resource result)

`mssql_fetch_object` retourne un objet dont les propriétés contiennent les valeurs de la ligne courante dans le résultat `result` , ou FALSE s'il n'y a plus de ligne.

`mssql_fetch_object` est similaire à `mssql_fetch_array` , avec une différence : un objet est retourné, au lieu d'un tableau. Indirectement, cela signifie que vous ne pouvez accéder aux données que par leur nom de champ, et pas par leur offset (les nombres sont illégaux comme nom de propriété).

Note
 Les noms des champs retournés par cette fonction sont **sensibles à la casse** .

Note

Cette fonction définit les champs NULL à la valeur PHP NULL .

En terme de vitesse, cette fonction est identique à [mssql_fetch_array](#) , quasiment aussi rapide que [mssql_fetch_row](#) (la différence est non significative).

Voir aussi [mssql_fetch_array](#) et [mssql_fetch_row](#) .

8.78.17 [mssql_fetch_row\(\)](#) : Lit une ligne de résultat MS SQL dans un tableau numérique

array [mssql_fetch_row](#) (resource result)

[mssql_fetch_row](#) retourne un tableau qui contient les valeurs de la ligne courante du résultat result ou bien FALSE s'il n'y a plus de ligne à lire.

[mssql_fetch_row](#) lit une ligne dans le résultat result et place les valeurs dans un tableau. Chaque valeur est enregistrée dans un élément du tableau, et les indices commencent à 0.

Les appels suivants à [mssql_fetch_row](#) retourneront la ligne suivante, ou bien FALSE s'il ne reste plus de ligne.

Note

Cette fonction définit les champs NULL à la valeur PHP NULL .

Voir aussi [mssql_fetch_array](#) , [mssql_fetch_object](#) , [mssql_data_seek](#) , [mssql_fetch_lengths](#) et [mssql_result](#) .

8.78.18 [mssql_field_length\(\)](#) : Lit la longueur d'un champ MS SQL Server

int [mssql_field_length](#) (resource result , int offset)

[mssql_field_length](#) retourne la longueur d'un champ numéro offset dans le résultat result . Si offset est omis, le champ courant est utilisé.

Note

Note pour les utilisateurs Windows

A cause d'une limitation dans l'API utilisé par PHP (MS DbLib C API), la longueur des champs VARCHAR est limitée à **255** . Si vous avez besoin d'enregistrer plus de données, utilisez un champ TEXT à la place.

8.78.19 [mssql_field_name\(\)](#) : Lit le nom d'un champ MS SQL Server

string [mssql_field_name](#) (resource result , int offset)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.78.20 `mssql_field_seek()` : Fixe l'offset du pointeur de champ MS SQL Server

bool `mssql_field_seek` (resource result , int field_offset)

`mssql_field_seek` modifie la valeur du pointeur de champ. Lors du prochain appel à `mssql_fetch_field` qui ne précisera pas de numéro de champ, le champ fixé par `mssql_field_seek` sera retourné.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `mssql_fetch_field` .

8.78.21 `mssql_field_type()` : Lit le nom d'un champ MS SQL Server

string `mssql_field_type` (resource result , int offset)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.78.22 `mssql_free_result()` : Libère la mémoire des ressources MS SQL Server

bool `mssql_free_result` (resource result)

`mssql_free_result` n'a besoin d'être appelé que si on craint d'utiliser trop de mémoire durant une opération. Toutes les ressources liées à un résultat seront libérées par `mssql_free_result` .

8.78.23 `mssql_free_statement()` : Libère une commande MS SQL Server de la mémoire

bool `mssql_free_statement` (resource statement)

`mssql_free_statement` n'est utile que si vous craignez de consommer trop de mémoire durant l'exécution de votre script. Toute la mémoire consommée par une commande sera libérée lorsque le script se termine. Avec la fonction `mssql_free_statement` , la mémoire associée au résultat statement sera libérée.

Voir aussi `mssql_bind` , `mssql_execute` et `mssql_init` .

8.78.24 `mssql_get_last_message()` : Retourne le dernier message d'erreur du serveur

string `mssql_get_last_message` (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.78.25 `mssql_guid_string()` : Convertit le GUID binaire de 16 octets en une chaîne de caractères

string `mssql_guid_string` (string `binary` , int `short_format`)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.78.26 `mssql_init()` : Initialise une procédure stockée MS SQL Server locale ou distante

resource `mssql_init` (string `sp_name` , resource `conn_id`)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Voir aussi [mssql_bind](#) , [mssql_execute](#) et [mssql_free_statement](#)

8.78.27 `mssql_min_error_severity()` : Fixe le niveau de sévérité des erreurs MS SQL Server

void `mssql_min_error_severity` (int `severity`)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.78.28 `mssql_min_message_severity()` : Fixe le niveau de sévérité des messages d'erreurs MS SQL Server

void `mssql_min_message_severity` (int `severity`)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.78.29 `mssql_next_result()` : Déplace le pointeur interne MS SQL Server au résultat suivant

bool `mssql_next_result` (resource `result_id`)

Lorsque vous envoyez plus d'une commande SQL au serveur, ou que vous exécutez une procédure stockée avec de multiples résultats, cela va conduire le serveur à retourner plusieurs jeux de lignes. `mssql_next_result` va vérifier l'existence de plusieurs résultats disponibles sur le serveur. Si un autre jeu de résultat existe, `mssql_next_result` va détruire de résultat précédent et préparer la lecture dans les nouvelles lignes.

`mssql_next_result` retourne TRUE si un autre résultat est disponible, ou FALSE sinon.

Exemple avec `mssql_next_result`

```
<?php
    $link = mssql_connect("localhost", "userid", "secret");
```

```
mssql_select_db("MyDB", $link);
$SQL = "Select * from table1 select * from table2";
$rs = mssql_query($SQL, $link);
do {
    while ($row = mssql_fetch_row($rs)) {
    }
} while (mssql_next_result($rs));
mssql_free_result($rs);
mssql_close($link);
?>
```

8.78.30 **mssql_num_fields()** : Retourne le nombre de champs dans un résultat MS SQL Server

int **mssql_num_fields** (resource result)

mssql_num_fields retourne le nombre de champs dans un résultat.

Voir aussi mssql_query , mssql_fetch_field et mssql_num_rows .

8.78.31 **mssql_num_rows()** : Retourne le nombre de lignes dans un résultat MS SQL

int **mssql_num_rows** (resource result)

mssql_num_rows retourne le nombre de lignes dans le résultat result .

Voir aussi mssql_query et mssql_fetch_row .

8.78.32 **mssql_pconnect()** : Ouvre une connexion persistante à un serveur MS SQL

resource **mssql_pconnect** (*string servername* , *string username* , *string password*)

mssql_pconnect retourne un identifiant positif de lien MS SQL en cas de succès, et FALSE en cas d'erreur.

mssql_pconnect se comporte comme mssql_connect mais avec deux différences :

Premièrement, lors de la connexion, la fonction va commencer par rechercher un lien persistant déjà ouvert avec le même hôte, le même nom d'utilisateur, username et le même mot de passe password . Si un tel lien est trouvé, cet identifiant sera retourné, au lieu d'ouvrir une autre connexion.

Deuxièmement, la connexion au serveur SQL ne sera pas fermée à la fin du script, mais restera ouverte, pour d'autres utilisations ultérieures (mssql_close ne fermera pas un lien établi avec mssql_pconnect).

C'est pourquoi ce type de lien est dit 'persistant'.

8.78.33 `mssql_query()` : Envoie une requête SQL au serveur MS SQL

mixed `mssql_query` (string `query` , *resource* `link_identifieur` , *int* `batch_size`)

`mssql_query` retourne un identifiant de résultat en cas de succès, TRUE si aucune ligne n'est retournée ou FALSE en cas d'échec.

`mssql_query` envoie la requête au serveur courant, associé à l'identifiant `link_identifieur` (ou la base par défaut, s'il est omis). Si aucun lien n'est ouvert, `mssql_query` essaiera d'ouvrir une connexion, en appelant `mssql_connect` .

Voir aussi `mssql_select_db` et `mssql_connect` .

8.78.34 `mssql_result()` : Lit les données d'un résultat

string `mssql_result` (*resource* `result` , *int* `row` , mixed `field`)

`mssql_result` retourne la valeur de la colonne, à la ligne donnée, dans le résultat MS SQL, ou FALSE en cas d'erreur.

`mssql_result` retourne le contenu d'une des cellules d'un résultat MS SQL. Le nom du champ peut être son nom littéral ou son offset, le nom de la table + "." + le nom du champ, ou la même chose avec le nom de la base de données. Si la colonne a été aliasée, utilisez le nom de l'alias plutôt que celui de la colonne.

Lorsque vous travaillez sur des résultats de grande taille, il vaut mieux utiliser les fonctions qui récupèrent toute une ligne (voir ci-après). Comme ces fonctions lisent toutes les valeurs en une passe, elles sont EXTREMEMENT PLUS RAPIDES que `mssql_result` . De plus, pensez que l'utilisation de l'offset numérique est beaucoup plus rapide que l'utilisation du nom de la colonne.

Alternatives recommandées : `mssql_fetch_row` , `mssql_fetch_array` et `mssql_fetch_object` .

8.78.35 `mssql_rows_affected()` : Retourne le nombre de lignes affectées par une requête MS SQL Server

int `mssql_rows_affected` (*resource* `conn_id`)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.78.36 `mssql_select_db()` : Sélectionne la base de données MS SQL

bool `mssql_select_db` (string `database_name` , *resource* `link_identifieur`)

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

`mssql_select_db` sélectionne la base de données active. Si aucun identifiant de connexion n'est fourni, la fonction utilisera la dernière connexion ouverte. Si aucune n'a été ouverte, la fonction essaiera d'en ouvrir une avec `mssql_connect` , et de l'utiliser.

Tous les appels à `mssql_query` seront faits dans cette base.

Pour échapper le nom de la base de données qui contient des espaces, des tirets ("-") ou tous autres caractères spéciaux, le nom de la base de données doit être entouré de crochets ([]), comme montré dans l'exemple ci-dessous. Cette technique peut également être appliquée lors de la sélection d'un nom de base de données qui est aussi un mot réservé (comme "primary").

Exemple avec [mysql_select_db](#)

```
<?php
    $conn = mysql_connect('MYSQLSERVER', 'sa', 'password');
    mysql_select_db('[my data-base]', $conn);
?>
```

Voir aussi [mysql_connect](#) , [mysql_pconnect](#) et [mysql_query](#) .

8.79 Fonctions Muscat

8.79.1 Introduction

Attention

Ce module est **EXPERIMENTAL** . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, **SANS PREAVIS!** Soyez-en conscient, et utilisez ce module à vos risques et périls.

8.79.2 Installation

Ces fonctions ne sont disponibles que si PHP a été compilé avec `--with-muscat[=DIR]` .

- [Introduction](#)
- [Installation](#)
- [muscat_close](#)
- [muscat_get](#)
- [muscat_give](#)
- [muscat_setup_net](#)
- [muscat_setup](#)

8.79.4 muscat_get() : Lit la réponse de l'API muscat

string **muscat_get** (resource muscat_handle)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, **SANS PREAVIS!** Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Attention

Cette fonction peut retourner FALSE , mais elle peut aussi retourner une valeur équivalent à FALSE , utilisable dans une condition if simple. Utilisez l' opérateur === pour tester la valeur de retour exacte de cette fonction.

8.79.5 muscat_give() : Envoie une chaîne à l'API de base muscat

void **muscat_give** (resource muscat_handle , string string)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, **SANS PREAVIS!** Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.79.6 muscat_setup_net() : Crée une nouvelle session muscat

resource **muscat_setup_net** (string muscat_host)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

muscat_setup_net crée une nouvelle session muscat, et retourne une ressource de connexion. muscat_host est le nom de l'hôte et port est le numéro de port où se connecter.

8.79.7 muscat_setup() : Crée une nouvelle session

resource **muscat_setup** (int size , string muscat_dir)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

size est la quantité de mémoire allouée, en octets. muscat_dir est le dossier d'installation de muscat, c'est-à-dire "/usr/local/empower" , si vous ne le fournissez pas, la valeur sera celle du dossier de compilation de muscat.

8.80 MySQL

8.80.1 Introduction

Ces fonctions vous permettent d'accéder aux bases de données MySQL. Le site officiel de cette base est <http://www.mysql.com/> .

La documentation de MySQL est disponible à <http://dev.mysql.com/doc/> .

8.80.2 Pré-requis

Afin de pouvoir les utiliser, vous devez compiler PHP avec le support MySQL.

8.80.3 Installation

Pour compiler, utilisez tout simplement l'option de configuration `--with-mysql[=DIR]` où le paramètre optionnel [DIR] pointe vers le dossier d'installation de MySQL.

Cette extension MySQL ne supporte pas toutes les fonctionnalités des versions de MySQL supérieures à la version 4.1.0. Pour cela, utilisez l'extension [MySQLi](#) .

Si vous voulez installer l'extension `mysqli` en même temps que l'extension `mysql`, vous devez utiliser la même bibliothèque client afin d'éviter les conflits.

8.80.3.2 Installation sur les systèmes Linux

PHP 4

L'option `--with-mysql` est activée par défaut. Ce comportement par défaut peut être désactivé avec l'option de configuration `--without-mysql` . Si MySQL est activé sans spécifier le chemin vers le dossier d'installation de MySQL, PHP utilisera la bibliothèque cliente MySQL fournie dans la distribution PHP.

Les utilisateurs qui se servent d'autres applications qui utilisent MySQL (par exemple, `auth-mysql`) ne doivent pas utiliser la bibliothèque intégrée, mais plutôt spécifier `--with-mysql=/path/to/mysql` . Cela va forcer PHP à utiliser les bibliothèques clientes installées par MySQL, et évitera les conflits.

PHP 5+

MySQL n'est plus activé par défaut, et la bibliothèque MySQL n'est plus fournie avec PHP. Lisez la [FAQ](#) pour savoir pourquoi. Utilisez l'option de configuration `--with-mysql[=DIR]` pour inclure le support MySQL. Vous pouvez télécharger **les en-têtes et les bibliothèques** depuis le site officiel de [MySQL](#) .

8.80.3.3 Installation sur les systèmes Windows

PHP 4

L'extension MySQL est directement fournie dans la distribution PHP.

PHP 5+

MySQL n'est plus activé par défaut, ainsi, la bibliothèque `php_mysql.dll` doit être activée dans le `php.ini`. De plus, PHP doit avoir accès à la bibliothèque cliente MySQL. Un fichier nommé `libmysql.dll` est inclu dans la distribution de PHP pour Windows et pour que PHP puisse discuter avec MySQL, ce fichier doit être disponible dans le **PATH** du système Windows. Lisez la FAQ intitulée "[Où dois-je ajouter mon répertoire PHP à la variable PATH sous Windows ?](#)" pour plus d'informations sur la réalisation de cela. Néanmoins, le fait de copier le fichier `libmysql.dll` dans le répertoire système de Windows fonctionne (car le dossier système est par défaut dans le **PATH** système), mais cela n'est pas du tout recommandé.

Pour activer n'importe quelle extension PHP (comme `php_mysql.dll`), la directive PHP `extension_dir` doit être définie et doit pointer vers le dossier où sont stockées les extensions PHP. Lisez également le [manuel d'installation sous Windows](#). Par exemple, voici une valeur possible pour la directive `extension_dir` en PHP 5 : `c:\php\ext`

Note
Si lorsque vous démarrez le serveur web une erreur similaire à ceci apparaît : "Unable to load dynamic library './php_mysql.dll'", c'est parce que <code>php_mysql.dll</code> et/ou <code>libmysql.dll</code> n'ont pu être trouvés par le système.

8.80.3.4 Notes sur l'installation de MySQL

Attention
Des crashes et des problèmes de démarrage de PHP peuvent être rencontrés lorsque vous chargez cette fonction en même temps que l'extension <code>recode</code> . Voyez l'extension <code>recode</code> pour plus de détails.
Note
Si vous avez besoin d'autres jeux de caractères que celui par défaut (latin), vous devez installer la bibliothèque externe <code>libmysql</code> (non fournie), compilée avec ce jeu de caractères.

8.80.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable	Historique
<code>mysql.allow_persistent</code>	"1"	PHP_INI_SYSTEM	
<code>mysql.max_persistent</code>	"-1"	PHP_INI_SYSTEM	
<code>mysql.max_links</code>	"-1"	PHP_INI_SYSTEM	
<code>mysql.trace_mode</code>	"Off"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
<code>mysql.default_port</code>	NULL	PHP_INI_ALL	
<code>mysql.default_socket</code>	NULL	PHP_INI_ALL	Disponible depuis PHP 4.0.1.
<code>mysql.default_host</code>	NULL	PHP_INI_ALL	
<code>mysql.default_user</code>	NULL	PHP_INI_ALL	
<code>mysql.default_password</code>	NULL	PHP_INI_ALL	
<code>mysql.connect_timeout</code>	"60"	PHP_INI_ALL	PHP_INI_SYSTEM en PHP <= 4.3.2. Disponible depuis PHP 4.3.0.

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

`mysql.allow_persistent` bool

Active ou désactive les connexions persistantes à la base de données MySQL.

`mysql.max_persistent` entier

Le nombre maximum de connexions persistantes aux bases de données MySQL, par processus.

`mysql.max_links` entier

Le nombre maximum de connexions aux bases de données MySQL, y compris les connexions persistantes, par processus.

`mysql.trace_mode` boolean

Mode trace. Lorsque `mysql.trace_mode` est activée, des alertes pour scans de table ou d'index, pour oubli de libération des ressources ou erreurs SQL seront affichées. Cette option a été introduite en PHP 4.3.0)

`mysql.default_port` string

Le numéro de port TCP par défaut, utilisé pour se connecter à la base de données, lorsqu'aucun port n'est spécifié. Si aucun port par défaut n'est spécifié, le port sera alors obtenu en lisant la variable d'environnement **MYSQL_TCP_PORT**, l'entrée `mysql-tcp` dans le fichier `/etc/services` ou encore la constante de compilation `MYSQL_PORT`, dans cet ordre. Windows n'utilisera aussi que la constante `MYSQL_PORT`.

`mysql.default_socket` string

Le nom par défaut de la socket lorsque l'on se connecte au serveur local, si aucune autre socket n'est spécifiée.

`mysql.default_host` string

Adresse par défaut du serveur, à utiliser lors de la connexion à un serveur MySQL, si aucun hôte n'est spécifié. Ne s'applique pas lorsque le safe mode SQL est activé.

`mysql.default_user` string

Utilisateur par défaut, à utiliser lors de la connexion à un serveur MySQL, si aucun utilisateur n'est spécifié. Ne s'applique pas lorsque le safe mode SQL est activé.

`mysql.default_password` string

Mot de passe par défaut, à utiliser lors de la connexion à un serveur MySQL, si aucun mot de passe n'est spécifié. Ne s'applique pas lorsque le safe mode SQL est activé.

`mysql.connect_timeout` entier

Durée maximale d'attente de la réponse d'un serveur, en secondes. Sous Linux, cette durée sert aussi lors de l'échange du premier avec le serveur.

8.80.5 Types de ressources

Il y a deux types de ressources utilisées par le module MySQL. Le premier est un identifiant de connexion au serveur, appelé `mysql link`, et le second est un identifiant de résultat de requête, appelé `mysql result`.

8.80.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Depuis PHP 4.3.0, il est possible de spécifier des options de client supplémentaires pour les fonctions `mysql_connect` et `mysql_pconnect`. Ces constantes sont les suivantes :

Constante	Description
MYSQL_CLIENT_COMPRESS	Utilise le protocole avec compression
MYSQL_CLIENT_IGNORE_SPACE	Autorise les espaces après les noms de fonction
MYSQL_CLIENT_INTERACTIVE	Autorise interactive_timeout secondes d'inactivité sur la connexion (au lieu de wait_timeout).
MYSQL_CLIENT_SSL	Utilisation du chiffrement SSL. Cette constante n'est disponible qu'à partir de la version 4.x et plus récente de la bibliothèque cliente MySQL. La version 3.23.x est fournis avec PHP 4 ainsi qu'avec les binaires pour windows de PHP 5.

La fonction `mysql_fetch_array` utilise une constante pour spécifier les différents types de formats de réponse. Les constantes suivantes sont utilisées :

Constante	Description
MYSQL_ASSOC	Les colonnes sont retournées dans un tableau, dont les index sont les noms de colonnes.
MYSQL_BOTH	Les colonnes sont retournées dans un tableau ayant une indexation numérique et un système d'index correspondant au nom des colonnes.
MYSQL_NUM	Les colonnes sont retournées dans un tableau avec un index numérique. Les colonnes sont numérotées dans leur ordre d'apparition. L'index commence à zéro.

8.80.7 Notes

Note
La plupart des fonctions MySQL acceptent <code>link_identifier</code> comme le dernier paramètre optionnel. S'il n'est pas fourni, la dernière connexion ouverte est utilisée. Si elle n'existe pas, une connexion est essayée d'être établie avec les paramètres par défaut définis dans <code>php.ini</code> . Si les fonctions ne réussissent pas, elles retournent <code>FALSE</code> .

8.80.8 Exemples

Cet exemple simple montre comment se connecter, exécuter une requête, lire les informations obtenues et se déconnecter d'une base de données MySQL.

Exemple de présentation de l'extension MySQL
<pre> <?php // Connexion et sélection de la base \$link = mysql_connect("mysql_hote", "mysql_utilisateur", "mysql_mot_de_passe") or die("Impossible de se connecter"); echo "Connexion réussie"; mysql_select_db("my_database") or die("Could not select database"); // Exécuter des requêtes SQL \$query = "SELECT * FROM my_table"; \$result = mysql_query(\$query) or die("Query failed"); // Afficher des résultats en HTML echo "<table>\n"; while (\$line = mysql_fetch_assoc(\$result)) { echo "\t<tr>\n"; foreach (\$line as \$col_value) { echo "\t\t<td>\$col_value</td>\n"; } echo "\t</tr>\n"; </pre>

```
}  
echo "</table>\n";  
  
// Libération des résultats  
mysql_free_result($result);  
  
// Fermeture de la connexion  
mysql_close($link);  
?>
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Notes](#)
- [Exemples](#)
- [mysql_affected_rows](#)
- [mysql_change_user](#)
- [mysql_client_encoding](#)
- [mysql_close](#)
- [mysql_connect](#)
- [mysql_create_db](#)
- [mysql_data_seek](#)
- [mysql_db_name](#)
- [mysql_db_query](#)
- [mysql_drop_db](#)
- [mysql_errno](#)
- [mysql_error](#)
- [mysql_escape_string](#)
- [mysql_fetch_array](#)
- [mysql_fetch_assoc](#)
- [mysql_fetch_field](#)
- [mysql_fetch_lengths](#)
- [mysql_fetch_object](#)
- [mysql_fetch_row](#)
- [mysql_field_flags](#)
- [mysql_field_len](#)
- [mysql_field_name](#)
- [mysql_field_seek](#)
- [mysql_field_table](#)
- [mysql_field_type](#)
- [mysql_free_result](#)
- [mysql_get_client_info](#)
- [mysql_get_host_info](#)
- [mysql_get_proto_info](#)
- [mysql_get_server_info](#)
- [mysql_info](#)
- [mysql_insert_id](#)
- [mysql_list_dbs](#)
- [mysql_list_fields](#)
- [mysql_list_processes](#)
- [mysql_list_tables](#)
- [mysql_num_fields](#)

- [mysql_num_rows](#)
- [mysql_pconnect](#)
- [mysql_ping](#)
- [mysql_query](#)
- [mysql_real_escape_string](#)
- [mysql_result](#)
- [mysql_select_db](#)
- [mysql_stat](#)
- [mysql_tablename](#)
- [mysql_thread_id](#)
- [mysql_unbuffered_query](#)

8.80.10 [mysql_change_user\(\)](#) : Change le nom de l'utilisateur de la connexion active

int **mysql_change_user** (string user , string password , string database , resource link_identifier)

[mysql_change_user](#) change l'utilisateur de la session courante, ou sur la connexion spécifiée avec l'option `link_identifier` . Le nouvel utilisateur est `user` , identifié par le mot de passe `password` . Si une base database est spécifiée, elle deviendra la base par défaut de l'utilisateur. Si une erreur de connexion survient, la connexion en cours restera active. Si `link_identifier` n'est pas précisé, [mysql_affected_rows](#) utilise la dernière connexion ouverte.

Cette fonction est dépréciée et n'existe plus dans PHP.

8.80.11 [mysql_client_encoding\(\)](#) : Retourne le nom du jeu de caractères utilisé par le client MySQL

string **mysql_client_encoding** (resource link_identifier)

Retourne la variable `character_set` de MySQL.

8.80.12 [mysql_close\(\)](#) : Ferme la connexion MySQL

bool **mysql_close** (resource link_identifier)

[mysql_close](#) ferme la connexion non persistante au serveur MySQL associée à l'identifiant spécifié. Si `link_identifier` n'est pas spécifié, cette commande s'applique à la dernière connexion ouverte.

L'utilisation de [mysql_close](#) n'est pas habituellement nécessaire, puisque les connexions non persistantes ouverts sont automatiquement fermées à la fin l'exécution du script. Voyez aussi [libérer des ressources](#) .

8.80.13 [mysql_connect\(\)](#) : Ouvre une connexion à un serveur MySQL

resource **mysql_connect** (string server , string username , string password , bool new_link , int client_flags)

Ouvre ou réutilise une connexion à un serveur MySQL.

8.80.14 `mysql_create_db()` : Crée une base de données MySQL

bool `mysql_create_db` (string `database_name` , *resource link_identifier*)

`mysql_create_db` tente de créer une nouvelle base de données sur le serveur associé avec l'identifiant de connexion spécifié.

8.80.15 `mysql_data_seek()` : Déplace le pointeur interne de résultat MySQL

bool `mysql_data_seek` (resource `result` , int `row_number`)

`mysql_data_seek` déplace le pointeur interne de résultat, dans le résultat associé à l'identifiant de résultat `result_identifier` . Il le fait pointer à la ligne `row_number` . Le prochain appel à `mysql_fetch_row` retournera cette ligne.

`row_number` commence à 0. `row_number` doit être une valeur qui va de 0 à `mysql_num_rows` - 1. Cependant, si le résultat est vide, un `row_number` de 0 échouera avec une erreur `E_WARNING` et `mysql_data_seek` retournera FALSE .

8.80.16 `mysql_db_name()` : Lit les noms des bases de données

string `mysql_db_name` (resource `result` , int `row` , *mixed field*)

Récupère le nom de base de données à l'appel de `mysql_list_dbs` .

8.80.17 `mysql_db_query()` : Envoie une requête MySQL à un serveur MySQL

resource `mysql_db_query` (string `database` , string `query` , *resource link_identifier*)

`mysql_db_query` sélectionne une base de données et exécute une requête sur celle-ci.

8.80.18 `mysql_drop_db()` : Efface une base de données MySQL

bool `mysql_drop_db` (string `database_name` , *resource link_identifier*)

`mysql_drop_db` essaie d'effacer la base de données `database_name` complète sur le serveur associé à la ressource de connexion `link_identifier` . La fonction `mysql_drop_db` est obsolète. Il est préférable d'utiliser `mysql_query` avec la requête DROP DATABASE à la place.

8.80.19 `mysql_errno()` : Retourne le numéro d'erreur de la dernière commande MySQL

int **mysql_errno** (*resource link_identifieur*)

Retourne le numéro d'erreur de la dernière commande MySQL.

Les erreurs qui sont remontées depuis le serveur MySQL ne sont plus des alertes. A la place, il faut utiliser mysql_errno pour obtenir le numéro d'erreur. Notez que cette fonction retourne uniquement le code erreur depuis la dernière fonction MySQL exécutée (n'incluant pas les fonctions mysql_error et mysql_errno), donc, si vous voulez l'utiliser, assurez-vous de récupérer la valeur avant d'appeler une autre fonction MySQL.

8.80.20 **mysql_error()** : Retourne le texte associé avec l'erreur générée lors de la dernière requête

string **mysql_error** (*resource link_identifieur*)

mysql_error retourne le message d'erreur généré par la dernière commande MySQL. Les erreurs retournées par le serveur MySQL ne génèrent plus de message d'alerte. A la place, vous devez utiliser la fonction mysql_error pour lire le contenu du message. Notez que cette fonction ne retourne que le texte de l'erreur la plus récente (n'incluant pas mysql_error et mysql_errno), ce qui fait que si vous souhaitez l'utiliser, vous devez vous assurer de sa valeur avant de lancer une autre requête.

8.80.21 **mysql_escape_string()** : Protège une chaîne pour la passer à mysql_query

string **mysql_escape_string** (string unescaped_string)

mysql_escape_string va protéger tous les caractères de la chaîne unescaped_string , pour pouvoir l'utiliser directement dans une requête mysql_query . Cette fonction est dépréciée.

mysql_escape_string est identique à la fonction mysql_real_escape_string hormis le fait que mysql_real_escape_string requiert une ressource de connexion et protège la chaîne en fonction du jeu de caractères courant. mysql_escape_string ne demande pas de connexion comme argument, et ne resecte pas le jeu de caractères courant.

8.80.22 **mysql_fetch_array()** : Retourne une ligne de résultat MySQL sous la forme d'un tableau associatif, d'un tableau indexé, ou les deux

array **mysql_fetch_array** (resource result , int result_type)

mysql_fetch_array retourne un tableau qui contient la ligne demandée dans le résultat result et déplace le pointeur de données interne d'un cran.

8.80.23 **mysql_fetch_assoc()** : Lit une ligne de résultat MySQL dans un tableau associatif

array **mysql_fetch_assoc** (resource result)

mysql_fetch_assoc retourne un tableau associatif qui contient la ligne lue dans le résultat `result` et déplace le pointeur interne de données. mysql_fetch_assoc revient à appeler la fonction mysql_fetch_array avec `MYSQL_ASSOC` en tant que second paramètre. Cette fonction retourne uniquement un tableau associatif.

8.80.24 mysql_fetch_field() : Retourne les données enregistrées dans une colonne MySQL sous forme d'objet

objet **mysql_fetch_field** (resource result , int field_offset)

Retourne un objet contenant les informations sur les champs. Cette fonction peut être utilisée pour obtenir des informations sur les champs de la requête fournie `result` .

8.80.25 mysql_fetch_lengths() : Retourne la taille de chaque colonne d'une ligne de résultat MySQL

array **mysql_fetch_lengths** (resource result)

mysql_fetch_lengths retourne un tableau avec la taille de chaque colonne de la dernière ligne retournée par mysql_fetch_row .

mysql_fetch_lengths stocke les tailles de chaque colonne de la dernière ligne retournée par mysql_fetch_row , mysql_fetch_assoc , mysql_fetch_array et mysql_fetch_object dans un tableau, en commençant à la position 0.

8.80.26 mysql_fetch_object() : Retourne une ligne de résultat MySQL sous la forme d'un objet

objet **mysql_fetch_object** (resource result)

mysql_fetch_object retourne un tableau qui contient la ligne demandée dans le résultat `result` et déplace le pointeur de données interne d'un cran.

8.80.27 mysql_fetch_row() : Retourne une ligne de résultat MySQL sous la forme d'un tableau

array **mysql_fetch_row** (resource result)

mysql_fetch_row retourne un tableau indexé numériquement, qui correspond à la prochaine ligne du résultat MySQL `result` et déplace le pointeur interne de données d'un cran.

8.80.28 mysql_field_flags() : Retourne des détails sur une colonne MySQL

string **mysql_field_flags** (resource result , int field_offset)

mysql_field_flags retourne le sémaphore associé au champ spécifié par `field_offset` , dans le résultat `result` . Les sémaphores sont retournés comme des mots, séparés par des espaces, ce qui les rend faciles à séparer, avec la commande explode .

8.80.29 mysql_field_len() : Retourne la taille d'un champ de résultat MySQL

int **mysql_field_len** (resource `result` , int `field_offset`)

mysql_field_len retourne la taille du champ spécifié.

8.80.30 mysql_field_name() : Retourne le nom d'une colonne dans un résultat MySQL

string **mysql_field_name** (resource `result` , int `field_offset`)

mysql_field_name retourne le nom du champ de l'index spécifié.

8.80.31 mysql_field_seek() : Déplace le pointeur de résultat vers une position donnée

bool **mysql_field_seek** (resource `result` , int `field_offset`)

mysql_field_seek place le pointeur de résultat `result` sur le champ spécifié par sa position `field_offset` . Lors du prochain appel à mysql_fetch_field qui n'aura pas d'argument d'index de champ, le champ désormais pointé sera retourné.

8.80.32 mysql_field_table() : Retourne le nom de la table MySQL où se trouve une colonne

string **mysql_field_table** (resource `result` , int `field_offset`)

mysql_field_table retourne le nom de la table où se trouve la colonne d'offset `field_offset` , dans le résultat MySQL `result` .

8.80.33 mysql_field_type() : Retourne le type d'une colonne MySQL spécifique

string **mysql_field_type** (resource `result` , int `field_offset`)

mysql_field_type est similaire à la fonction mysql_field_name . Les arguments sont identiques mais c'est le type de la colonne qui est retourné.

8.80.34 `mysql_free_result()` : Libère le résultat de la mémoire

`bool mysql_free_result (resource result)`

`mysql_free_result` libère toute la mémoire et les ressources utilisées par la ressource de résultat `result` .

`mysql_free_result` n'est à appeler que si vous avez peur d'utiliser trop de mémoire durant l'exécution de votre script. Toute la mémoire associée à l'identifiant de résultat sera automatiquement libérée.

8.80.35 `mysql_get_client_info()` : Lit les informations sur le client MySQL

`string mysql_get_client_info (void)`

`mysql_get_client_info` retourne une chaîne qui représente le numéro de version du client utilisé par PHP.

8.80.36 `mysql_get_host_info()` : Lit les informations sur l'hôte MySQL

`string mysql_get_host_info (resource link_identifieur)`

`mysql_get_host_info` retourne une chaîne qui représente le type de connexion utilisé avec la connexion `link_identifieur` , y compris le nom du serveur hôte.

8.80.37 `mysql_get_proto_info()` : Lit les informations sur le protocole MySQL

`int mysql_get_proto_info (resource link_identifieur)`

Lit les informations sur le protocole MySQL.

8.80.38 `mysql_get_server_info()` : Lit les informations sur le serveur MySQL

`string mysql_get_server_info (resource link_identifieur)`

Lit les informations sur le serveur MySQL.

8.80.39 `mysql_info()` : Lit des informations à propos de la dernière requête MySQL

`string mysql_info (resource link_identifieur)`

Lit des informations à propos de la dernière requête MySQL.

8.80.40 mysql_insert_id() : Retourne l'identifiant généré par la dernière requête INSERT MySQL

int **mysql_insert_id** (*resource link_identifieur*)

mysql_insert_id retourne le dernier identifiant généré par un champ de type AUTO_INCREMENT, sur la connexion MySQL courante ou sur la connexion spécifiée par link_identifieur .

8.80.41 mysql_list_dbs() : Liste les bases de données disponibles sur le serveur MySQL

resource **mysql_list_dbs** (*resource link_identifieur*)

mysql_list_dbs retournera un identifiant de résultat, qui contiendra les noms des bases de données disponibles sur la connexion MySQL courante ou sur la connexion spécifiée par link_identifieur .

8.80.42 mysql_list_fields() : Liste les champs d'une table MySQL

resource **mysql_list_fields** (string database_name , string table_name , *resource link_identifieur*)

Liste les champs de la table MySQL spécifiée.

La fonction mysql_list_fields est dépréciée. Il est préférable d'utiliser la fonction mysql_query avec comme requête SHOW COLUMNS FROM table [LIKE 'name'] à la place.

8.80.43 mysql_list_processes() : Liste les processus MySQL

resource **mysql_list_processes** (*resource link_identifieur*)

Liste les processus MySQL.

8.80.44 mysql_list_tables() : Liste les tables d'une base de données MySQL

resource **mysql_list_tables** (string database , *resource link_identifieur*)

Liste les tables d'une base de données MySQL spécifiée.

Cette fonction est dépréciée. Il est préférable d'utiliser la fonction mysql_query pour effectuer la requête SQL SHOW TABLES [FROM db_name] [LIKE 'pattern'] à la place.

8.80.45 mysql_num_fields() : Retourne le nombre de champs d'un résultat MySQL

int **mysql_num_fields** (resource result)

Retourne le nombre de champs d'une requête.

8.80.46 `mysql_num_rows()` : Retourne le nombre de lignes d'un résultat MySQL

int `mysql_num_rows` (*resource result*)

Récupère le nombre de lignes d'un jeu de résultat. Cette commande n'est disponible que pour les requêtes SELECT. Pour récupérer le nombre de lignes affectées par une requête INSERT, UPDATE ou DELETE, utilisez la fonction [mysql_affected_rows](#) .

8.80.47 `mysql_pconnect()` : Ouvre une connexion persistante à un serveur MySQL

resource mysql_pconnect (*string server* , *string username* , *string password* , *int client_flags*)

Ouvre une connexion persistante à un serveur MySQL.

[mysql_pconnect](#) se comporte exactement comme [mysql_connect](#) , mais avec deux différences majeures :

Premièrement, lors de la connexion, la fonction essaie de trouver une connexion permanente déjà ouverte sur cet hôte, avec le même nom d'utilisateur et de mot de passe. Si une telle connexion est trouvée, son identifiant est retourné, sans ouvrir de nouvelle connexion.

Deuxièmement, la connexion au serveur MySQL ne sera pas terminée avec la fin du script. Au lieu de cela, le lien sera conservé pour un prochain accès ([mysql_close](#) ne terminera pas une connexion persistante établie par [mysql_pconnect](#)).

C'est pourquoi ce type de connexion est dite 'persistante'.

8.80.48 `mysql_ping()` : Pingue le serveur MySQL, et s'y reconnecte au besoin

bool `mysql_ping` (*resource link_identifier*)

[mysql_ping](#) vérifie si la connexion au serveur `link_identifier` fonctionne. Si elle est tombée, une reconnexion automatique est effectuée. Cette fonction est utile pour les scripts qui restent inactifs durant un long moment. Ils peuvent ainsi vérifier si le serveur a fermé la connexion, pour cause d'inactivité.

8.80.49 `mysql_query()` : Envoi une requête à un serveur MySQL

resource mysql_query (*string query* , *resource link_identifier*)

[mysql_query](#) envoie une requête (au serveur MySQL actuellement actif sur le serveur associé avec le `link_identifier` spécifié).

8.80.50 `mysql_real_escape_string()` : Protège les caractères spéciaux d'une commande SQL

string `mysql_real_escape_string` (string `unescaped_string` , *resource* `link_identifier`)

`mysql_real_escape_string` protège les caractères spéciaux de la chaîne `unescaped_string` , en prenant en compte le jeu de caractères courant de la connexion `link_identifier` . Le résultat peut être utilisé sans problème avec la fonction `mysql_query` . Si des données binaires doivent être insérées, cette fonction doit être utilisée.

`mysql_real_escape_string` appelle la fonction `mysql_escape_string` de la bibliothèque MySQL qui ajoute un slash aux caractères suivants : NULL , \x00 , \n , \r , \ , ' , " et \x1a .

Cette fonction doit toujours (avec quelques exceptions) être utilisée pour protéger vos données avant d'envoyer la requête à MySQL.

8.80.51 `mysql_result()` : Retourne un champ d'un résultat MySQL

string `mysql_result` (*resource* `result` , int `row` , *mixed* `field`)

Retourne le contenu d'un champ d'un jeu de résultats MySQL.

Lorsque vous travaillez sur des résultats de grande taille, il est conseillé d'utiliser une des fonctions qui vont rechercher une ligne entière dans un tableau. Ces fonctions sont NETTEMENT plus rapides. De plus, utiliser un offset numérique est aussi beaucoup plus rapide que spécifier un nom littéral.

8.80.52 `mysql_select_db()` : Sélectionne une base de données MySQL

bool `mysql_select_db` (string `database_name` , *resource* `link_identifier`)

Sélectionne une base de données MySQL sur le serveur associé avec le paramètre `link_identifier` . Chaque appel à la fonction `mysql_query` sera exécutée sur la base de données active.

8.80.53 `mysql_stat()` : Retourne le statut courant du serveur MySQL

string `mysql_stat` (*resource* `link_identifier`)

`mysql_stat` retourne le statut courant du serveur MySQL.

8.80.54 `mysql_tablename()` : Lit le nom de la table qui contient un champ

string `mysql_tablename` (*resource* `result` , int `i`)

Lit le nom de la table depuis le jeu de résultats `result` .

Cette fonction est dépréciée. Il est préférable d'utiliser la fonction `mysql_query` avec une requête SQL SHOW TABLES [FROM `db_name`] [LIKE '`pattern`'] à la place.

8.80.55 `mysql_thread_id()` : Retourne l'identifiant du thread MySQL courant

int `mysql_thread_id` (*resource link_identifieur*)

`mysql_thread_id` retourne l'identifiant du thread courant. Si la connexion `link_identifieur` est perdue et que vous vous reconnectez (avec `mysql_ping` , par exemple), alors l'identifiant de thread va changer. Cela signifie que vous devez le récupérer lorsque vous en avez besoin.

8.80.56 `mysql_unbuffered_query()` : Exécute une requête SQL sans mobiliser les résultats MySQL

resource `mysql_unbuffered_query` (string `query` , *resource link_identifieur*)

`mysql_unbuffered_query` envoie la requête SQL `query` au serveur MySQL identifié par `link_identifieur` , sans préparer les résultats pour la lecture, comme le fait `mysql_query` . D'une part, cela réduit considérablement la consommation de mémoire par MySQL, lorsque les requêtes génèrent des résultats de grande taille. D'autre part, vous pourrez utiliser les résultats dès que la première ligne aura été lue : pas besoin d'attendre que la requête ait complètement été exécutée. Lorsque vous utilisez de multiples connexions à MySQL, vous devez spécifier le paramètre optionnel `link_identifieur` .

8.81 Extension MySQL améliorée

8.81.1 Introduction

L'extension `mysqli` vous permet d'accéder aux fonctionnalités fournies par MySQL 4.1 et plus. Plus d'informations sur le serveur de bases de données MySQL peuvent être trouvées sur <http://www.mysql.com/>

La documentation de MySQL peut être trouvée sur <http://dev.mysql.com/doc/> .

Une partie de cette documentation provient du manuel officiel MySQL avec la permission de MySQL AB.

8.81.2 Pré-requis

Pour faire fonctionner ces fonctions, vous devez compiler PHP avec le support de l'extension MySQLi.

Note
L'extension MySQLi est conçue pour fonctionner avec la version 4.1.3 ou plus de MySQL. Pour les versions plus anciennes, voyez la documentation de l'extension MySQL .

8.81.3 Installation

Pour installer l'extension `mysqli` pour PHP, utilisez l'option de configuration `--with-mysqli=mysql_config_path` où `mysql_config_path` représente l'endroit où se trouve le programme `mysql_config` qui accompagne les versions de MySQL supérieures à la 4.1. Désactivez aussi l'extension standard [MySQL](#) (qui est activée par défaut) en utilisant aussi l'option de configuration `--without-mysql` . Si vous voulez installer l'extension standard `mysql` en même temps que l'extension `mysqli`, la bibliothèque embarquée `libmysql` distribuée avec PHP ne peut être utilisée. Utilisez à la place les bibliothèques clientes installées par MySQL avec des versions inférieures à la 4.1. Cela forcera PHP à utiliser les bibliothèques du client installées par MySQL pour éviter les conflits.

8.81.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable	Historique
<code>mysqli.max_links</code>	"-1"	PHP_INI_SYSTEM	Disponible depuis PHP 5.0.0.
<code>mysqli.default_port</code>	NULL	PHP_INI_ALL	Disponible depuis PHP 5.0.0.
<code>mysqli.default_socket</code>	NULL	PHP_INI_ALL	Disponible depuis PHP 5.0.0.
<code>mysqli.default_host</code>	NULL	PHP_INI_ALL	Disponible depuis PHP 5.0.0.
<code>mysqli.default_user</code>	NULL	PHP_INI_ALL	Disponible depuis PHP 5.0.0.
<code>mysqli.default_pw</code>	NULL	PHP_INI_ALL	Disponible depuis PHP 5.0.0.

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

`mysqli.max_links` entier

Le nombre maximal de connexions MySQL par processus, incluant les connexions persistantes.

`mysqli.default_port` string

Le numéro par défaut de port TCP à utiliser lors de la connexion au serveur si aucun autre port n'est fourni. S'il n'est pas défini, le port sera obtenu à partir de la variable d'environnement `MYSQL_TCP_PORT`, l'entrée `mysql-tcp` dans `/etc/services` ou la constante de compilation `MYSQL_PORT`, dans cet ordre. Win32 n'utilisera que la constante `MYSQL_PORT`.

`mysqli.default_socket` string

Le nom par défaut de la socket à utiliser lors des connexions locales au serveur si aucun autre nom n'est fourni.

`mysqli.default_host` string

Le serveur par défaut à utiliser lors de la connexion à un serveur si aucun autre hôte n'est fourni. Ne s'applique pas avec le safe mode.

`mysqli.default_user` string

Le nom d'utilisateur par défaut à utiliser lors de la connexion à un serveur si aucun autre nom n'est fourni. Ne s'applique pas avec le safe mode.

`mysqli.default_pw` string

Le mot de passe par défaut à utiliser lors de la connexion à un serveur si aucun autre mot de passe n'est fourni. Ne s'applique pas avec le safe mode.

8.81.5 Classes pré-définies

8.81.5.1 `mysqli`

Représente la connexion entre PHP et le serveur MySQL.

8.81.5.1.1 Constructeur

- `mysqli` - construit un nouvel objet `mysqli`

8.81.5.1.2 Méthodes

- `autocommit` - Active ou désactive l'auto-commit
- `change_user` - Change l'utilisateur de la connexion
- `character_set_name` - Retourne le jeu de caractères par défaut de la connexion
- `close` - Ferme une connexion ouverte
- `commit` - Valide la transaction courante
- `connect` - Ouvre une nouvelle connexion au serveur MySQL
- `debug` - Effectue des opérations de débogage
- `dump_debug_info` - Affiche les informations de débogage
- `get_client_info` - Retourne la version du client
- `get_host_info` - Retourne le type de connexion utilisé
- `get_server_info` - Retourne la version du serveur MySQL
- `get_server_version` - Retourne la version du serveur MySQL
- `init` - Initialise un objet `mysqli`
- `info` - Lit les informations concernant la dernière requête
- `kill` - Tente de terminer un thread serveur
- `multi_query` - Exécute une requête multiple
- `more_results` - Vérifie si d'autres résultats existent après une commande multiple

- next_result - Lit le prochain résultat d'une commande multiple
- options - Modifie des options
- ping - Ping la connexion au serveur ou se reconnecte si elle est morte
- prepare - Prépare une requête SQL
- query - Exécute une requête SQL
- real_connect - Tente de se connecter au serveur MySQL
- escape_string - Protège les caractères spéciaux dans une chaîne
- rollback - Annule la transaction courante
- select_db - Sélectionne la base de données par défaut
- set_charset - Définit le jeu de caractères par défaut du client
- ssl_set - Configure les paramètres SSL
- stat - Lit le statut courant du système
- stmt_init - Initialise une commande pour utilisation avec mysqli_stmt_prepare
- store_result - Lit le jeu de résultat de la dernière requête
- use_result - Transfère un résultat non-bufferisé
- thread_safe - Indique si la sécurité thread est activée

8.81.5.1.3 Propriétés

- affected_rows - Lit le nombre de lignes affectées par la dernière opération
- client_info - Retourne la version cliente de MySQL en tant que chaîne de caractères
- client_version - Retourne la version cliente de MySQL en tant qu'entier
- errno - Retourne le code d'erreur du dernier appel de fonction
- error - Retourne le message d'erreur du dernier appel de fonction
- field_count - Retourne le nombre de colonnes de la dernière commande
- host_info - Retourne une chaîne décrivant le type de connexion utilisé
- info - Lit les informations sur la dernière commande exécutée
- insert_id - Retourne le dernier identifiant généré
- protocol_version - Retourne la version du protocole MySQL utilisé
- server_info - Retourne une chaîne de caractères qui représente le numéro de version du serveur
- server_version - Retourne le numéro de version du serveur en tant qu'entier
- sqlstate - Retourne une chaîne contenant le code d'erreur SQLSTATE
- thread_id - Retourne l'identifiant de thread de la connexion courante
- warning_count - Retourne le nombre d'alertes générées durant l'exécution de la dernière commande SQL

8.81.5.2 mysqli_stmt

Représente une commande préparée.

8.81.5.2.1 Méthodes

- bind_param - Associe des variables à une commande préparée
- bind_result - Associe des variables à un résultat
- close - Termine une commande préparée
- data_seek - Place le curseur à une ligne arbitraire dans un jeu de résultat
- execute - Exécute une commande préparée
- fetch - Lit une ligne de résultat dans des variables associées
- free_result - Libère le résultat de la mémoire
- result_metadata - Lit un résultat d'une commande préparée
- prepare - Prépare une commande SQL

- send_long_data - Envoie les données par bloc
- reset - annule une commande SQL
- store_result - Bufferise complètement le résultat d'une commande préparée

8.81.5.2.2 Propriétés

- affected_rows - Retourne le nombre de lignes affectées par la dernière commande SQL
- errno - Retourne le code d'erreur de la dernière commande
- error - Retourne le message d'erreur de la dernière commande
- field_count - Retourne le nombre de colonnes dans un jeu de résultats
- id - Retourne l'identifiant de la requête
- insert_id - Retourne la valeur générée pour une colonne AUTO_INCREMENT par la requête préparée
- num_rows - Retourne le nombre de lignes dans le jeu de résultats
- param_count - Retourne le nombre de paramètres nécessaires pour une commande préparée
- sqlstate - Retourne une chaîne contenant le code d'erreur SQLSTATE de la dernière commande

8.81.5.3 mysqli_result

Représente le résultat retourné par le serveur.

8.81.5.3.1 Méthodes

- close - Termine le jeu de résultat MySQL
- data_seek - Déplace le pointeur de lignes de résultat
- fetch_field - Lit les informations de colonnes dans un résultat
- fetch_fields - Lit les informations de toutes les colonnes d'un résultat
- fetch_field_direct - Lit les informations de colonne
- fetch_array - Lit une ligne de résultat dans un tableau associatif ou numérique
- fetch_assoc - Lit une ligne de résultat dans un tableau associatif
- fetch_object - Lit une ligne de résultat dans un objet
- fetch_row - Lit une ligne de résultat dans un tableau numérique
- close - Libère le résultat de la mémoire
- field_seek - Place le pointeur de résultat à un offset valide

8.81.5.3.2 Propriétés

- current_field - Retourne la position du pointeur de résultat
- field_count - Retourne le nombre de champs du résultat
- lengths - Retourne un tableau avec les tailles de colonnes
- num_rows - Retourne le nombre de lignes du résultat
- type - Retourne MYSQLI_STORE_RESULT ou MYSQLI_USE_RESULT

8.81.6 Constantes pré-définies

Nom	Description
MYSQLI_READ_DEFAULT_GROUP	Lit les options dans le groupe `my.cnf' ou dans le fichier

(entier)	spécifié par MYSQLI_READ_DEFAULT_FILE
MYSQLI_READ_DEFAULT_FILE (entier)	Lit les options dans le fichier spécifié, plutôt que dans my.cnf
MYSQLI_OPT_CONNECT_TIMEOUT (entier)	Durée d'expiration de la connexion, en secondes
MYSQLI_OPT_LOCAL_INFILE (entier)	Active la commande LOAD LOCAL INFILE
MYSQLI_INIT_COMMAND (entier)	Commande à exécuter lors de la connexion au serveur MySQL. Cette commande sera exécutée automatiquement lors de la reconnexion au serveur.
MYSQLI_CLIENT_SSL (entier)	Utilise le protocole SSL (chiffrement). Cette option ne doit pas être activée par un programme : elle doit être activée en interne, par la bibliothèque MySQL.
MYSQLI_CLIENT_COMPRESS (entier)	Utilise le protocole compressé.
MYSQLI_CLIENT_INTERACTIVE (entier)	Permet interactive_timeout secondes (au lieu de wait_timeout secondes) d'inactivité avant de fermer la connexion. La valeur de la variable wait_timeout du client prendra la valeur de interactive_timeout .
MYSQLI_CLIENT_IGNORE_SPACE (entier)	Permet les espaces après un nom de fonction. Cela fait de tous les noms de fonctions, des mots réservés.
MYSQLI_CLIENT_NO_SCHEMA (entier)	Interdit la syntaxe db_name.tbl_name.col_name.
MYSQLI_CLIENT_MULTI_QUERIES (entier)	
MYSQLI_STORE_RESULT (entier)	Pour les résultats mis en tampon.
MYSQLI_USE_RESULT (entier)	Pour les résultats non-mis en tampon.
MYSQLI_ASSOC (entier)	Les colonnes sont retournées dans le tableau, avec leurs noms comme index.
MYSQLI_NUM (entier)	Les colonnes sont retournées dans le tableau, avec leurs numéros comme index.
MYSQLI_BOTH (entier)	Les colonnes sont retournées dans le tableau, avec leurs noms et leurs numéros comme index.
MYSQLI_NOT_NULL_FLAG (entier)	Indique qu'un champ est défini comme NOT NULL
MYSQLI_PRI_KEY_FLAG (entier)	Le champ est une clé primaire.
MYSQLI_UNIQUE_KEY_FLAG (entier)	Le champ est un index unique.
MYSQLI_MULTIPLE_KEY_FLAG (entier)	Le champ fait partie d'un index.
MYSQLI_BLOB_FLAG (entier)	Le champ est de type BLOB
MYSQLI_UNSIGNED_FLAG (entier)	Le champ est de type UNSIGNED
MYSQLI_ZEROFILL_FLAG (entier)	Le champ est de type ZEROFILL
MYSQLI_AUTO_INCREMENT_FLAG (entier)	Le champ est de type AUTO_INCREMENT
MYSQLI_TIMESTAMP_FLAG (entier)	Le champ est de type TIMESTAMP
MYSQLI_SET_FLAG (entier)	Le champ est de type SET
MYSQLI_NUM_FLAG (entier)	Le champ est de type NUMERIC
MYSQLI_PART_KEY_FLAG (entier)	Le champ fait partie d'un index multiple
MYSQLI_GROUP_FLAG (entier)	Le champ fait partie de la clause GROUP BY
MYSQLI_TYPE_DECIMAL (entier)	Le champ est de type DECIMAL

MYSQLI_TYPE_TINY (<u>entier</u>)	Le champ est de type TINYINT
MYSQLI_TYPE_SHORT (<u>entier</u>)	Le champ est de type INT
MYSQLI_TYPE_LONG (<u>entier</u>)	Le champ est de type INT
MYSQLI_TYPE_FLOAT (<u>entier</u>)	Le champ est de type FLOAT
MYSQLI_TYPE_DOUBLE (<u>entier</u>)	Le champ est de type DOUBLE
MYSQLI_TYPE_NULL (<u>entier</u>)	Le champ est de type DEFAULT NULL
MYSQLI_TYPE_TIMESTAMP (<u>entier</u>)	Le champ est de type TIMESTAMP
MYSQLI_TYPE_LONGLONG (<u>entier</u>)	Le champ est de type BIGINT
MYSQLI_TYPE_INT24 (<u>entier</u>)	Le champ est de type MEDIUMINT
MYSQLI_TYPE_DATE (<u>entier</u>)	Le champ est de type DATE
MYSQLI_TYPE_TIME (<u>entier</u>)	Le champ est de type TIME
MYSQLI_TYPE_DATETIME (<u>entier</u>)	Le champ est de type DATETIME
MYSQLI_TYPE_YEAR (<u>entier</u>)	Le champ est de type YEAR
MYSQLI_TYPE_NEWDATE (<u>entier</u>)	Le champ est de type DATE
MYSQLI_TYPE_ENUM (<u>entier</u>)	Le champ est de type ENUM
MYSQLI_TYPE_SET (<u>entier</u>)	Le champ est de type SET
MYSQLI_TYPE_TINY_BLOB (<u>entier</u>)	Le champ est de type TINYBLOB
MYSQLI_TYPE_MEDIUM_BLOB (<u>entier</u>)	Le champ est de type MEDIUMBLOB
MYSQLI_TYPE_LONG_BLOB (<u>entier</u>)	Le champ est de type LONGBLOB
MYSQLI_TYPE_BLOB (<u>entier</u>)	Le champ est de type BLOB
MYSQLI_TYPE_VAR_STRING (<u>entier</u>)	Le champ est de type VARCHAR
MYSQLI_TYPE_STRING (<u>entier</u>)	Le champ est de type CHAR
MYSQLI_TYPE_GEOMETRY (<u>entier</u>)	Le champ est de type GEOMETRY
MYSQLI_NEED_DATA (<u>entier</u>)	Il reste des variables à relier
MYSQLI_NO_DATA (<u>entier</u>)	Il n'y a plus de variables à relier
MYSQLI_DATA_TRUNCATED (<u>integer</u>)	Les données ont été tronquées. Disponible depuis PHP 5.1.0 et MySQL 5.0.5.

Exemples

Tous les exemples dans la documentation MySQLi utilisent la base de données "world" provenant de MySQL AB. La base de données "world" peut être trouvée ici : <http://dev.mysql.com/get/Downloads/Manual/world.sql.gz/from/pick>

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Classes pré-définies](#)
- [Constantes pré-définies](#)
- [mysqli_affected_rows](#)
- [mysqli_autocommit](#)
- [mysqli_bind_param](#)
- [mysqli_bind_result](#)
- [mysqli_change_user](#)
- [mysqli_character_set_name](#)
- [mysqli_client_encoding](#)
- [mysqli_close](#)
- [mysqli_commit](#)

- [mysqli_connect_errno](#)
- [mysqli_connect_error](#)
- [mysqli_connect](#)
- [mysqli_data_seek](#)
- [mysqli_debug](#)
- [mysqli_disable_reads_from_master](#)
- [mysqli_disable_rpl_parse](#)
- [mysqli_dump_debug_info](#)
- [mysqli_embedded_connect](#)
- [mysqli_enable_reads_from_master](#)
- [mysqli_enable_rpl_parse](#)
- [mysqli_errno](#)
- [mysqli_error](#)
- [mysqli_escape_string](#)
- [mysqli_execute](#)
- [mysqli_fetch_array](#)
- [mysqli_fetch_assoc](#)
- [mysqli_fetch_field_direct](#)
- [mysqli_fetch_field](#)
- [mysqli_fetch_fields](#)
- [mysqli_fetch_lengths](#)
- [mysqli_fetch_object](#)
- [mysqli_fetch_row](#)
- [mysqli_fetch](#)
- [mysqli_field_count](#)
- [mysqli_field_seek](#)
- [mysqli_field_tell](#)
- [mysqli_free_result](#)
- [mysqli_get_client_info](#)
- [mysqli_get_client_version](#)
- [mysqli_get_host_info](#)
- [mysqli_get_metadata](#)
- [mysqli_get_proto_info](#)
- [mysqli_get_server_info](#)
- [mysqli_get_server_version](#)
- [mysqli_info](#)
- [mysqli_init](#)
- [mysqli_insert_id](#)
- [mysqli_kill](#)
- [mysqli_master_query](#)
- [mysqli_more_results](#)
- [mysqli_multi_query](#)
- [mysqli_next_result](#)
- [mysqli_num_fields](#)
- [mysqli_num_rows](#)
- [mysqli_options](#)
- [mysqli_param_count](#)
- [mysqli_ping](#)
- [mysqli_prepare](#)
- [mysqli_query](#)
- [mysqli_real_connect](#)
- [mysqli_real_escape_string](#)
- [mysqli_real_query](#)
- [mysqli_report](#)

- [mysqli_rollback](#)
- [mysqli_rpl_parse_enabled](#)
- [mysqli_rpl_probe](#)
- [mysqli_rpl_query_type](#)
- [mysqli_select_db](#)
- [mysqli_send_long_data](#)
- [mysqli_send_query](#)
- [mysqli_server_end](#)
- [mysqli_server_init](#)
- [mysqli_set_charset](#)
- [mysqli_set_opt](#)
- [mysqli_sqlstate](#)
- [mysqli_ssl_set](#)
- [mysqli_stat](#)
- [mysqli_stmt_affected_rows](#)
- [mysqli_stmt_bind_param](#)
- [mysqli_stmt_bind_result](#)
- [mysqli_stmt_close](#)
- [mysqli_stmt_data_seek](#)
- [mysqli_stmt_errno](#)
- [mysqli_stmt_error](#)
- [mysqli_stmt_execute](#)
- [mysqli_stmt_fetch](#)
- [mysqli_stmt_free_result](#)
- [mysqli_stmt_init](#)
- [mysqli_stmt_num_rows](#)
- [mysqli_stmt_param_count](#)
- [mysqli_stmt_prepare](#)
- [mysqli_stmt_reset](#)
- [mysqli_stmt_result_metadata](#)
- [mysqli_stmt_send_long_data](#)
- [mysqli_stmt_sqlstate](#)
- [mysqli_stmt_store_result](#)
- [mysqli_store_result](#)
- [mysqli_thread_id](#)
- [mysqli_thread_safe](#)
- [mysqli_use_result](#)
- [mysqli_warning_count](#)

8.81.8 [mysqli_autocommit\(\)](#) : [mysqli->autocommit\(\)](#)

Style procédural

bool **mysqli_autocommit** (mysqli link , bool mode)

Style orienté objet (méthode)

mysqli bool **autocommit** (bool mode)

[mysqli_autocommit](#) est utilisée pour activer ou désactiver le mode auto-commit pour les requêtes sur la connexion à la base de données représentée par le paramètre link .

Note

`mysqli_autocommit` ne fonctionne pas avec les types de tables non transactionnelles. (Comme MyISAM ou ISAM.)

Pour vérifier l'état de l'autocommit, utilisez la commande SQL 'SELECT @@autocommit'.

8.81.9 `mysqli_bind_param()` : Alias de `mysqli_stmt_bind_param`

`mysqli_bind_param` est un alias de la fonction `mysqli_stmt_bind_param`. Pour une description détaillée, voir la description de la fonction `mysqli_stmt_bind_param`.

Note

`mysqli_bind_param` est fortement déconseillée et devrait être supprimée prochainement.

8.81.10 `mysqli_bind_result()` : Alias de `mysqli_stmt_bind_result`

`mysqli_bind_result` est un alias de la fonction `mysqli_stmt_bind_result`. Pour une description détaillée, voir la description de la fonction `mysqli_stmt_bind_result`.

Note

`mysqli_bind_result` est fortement déconseillée et devrait être supprimée prochainement.

8.81.11 `mysqli_change_user()` : `mysqli->change_user()`

Style procédural

bool `mysqli_change_user` (mysqli link , string user , string password , string database)

Style orienté objet (méthode)

mysqli bool `change_user` (string user , string password , string database)

`mysqli_change_user` est utilisée pour changer l'utilisateur de la connexion spécifiée par le paramètre link et changer la base de données courante pour celle spécifiée par le paramètre database.

Vous pouvez passer la valeur NULL au paramètre database pour ne changer que l'utilisateur. Dans ce cas, vous pouvez utiliser la fonction `mysqli_select_db` pour changer de base de données.

Pour que cette fonction réussisse, les paramètres username et password doivent être valides et l'utilisateur en question doit avoir les permissions d'accès à la base de données désirée. Si pour une raison ou une autre, l'autorisation échoue, l'utilisateur courant sera conservé.

Note

L'utilisation de cette commande implique toujours que la connexion soit considérée comme neuve, que la fonction réussisse ou non. Un appel à cette fonction annulera donc toutes les transactions actives, fermera les tables temporaires et déverrouillera les tables verrouillées.

8.81.12 `mysqli_character_set_name()` : `mysqli->character_set_name()`

Style procédural

string `mysqli_character_set_name` (mysqli link)

Style orienté objet (méthode)

mysqli string **character_set_name** ()

Retourne le jeu de caractères courant pour la connexion spécifiée par le paramètre link .

8.81.13 **mysqli_client_encoding()** : Alias de **mysqli_character_set_name**

mysqli_client_encoding est un alias de la fonction mysqli_character_set_name . Pour une description détaillée, voir la description de la fonction mysqli_character_set_name .

8.81.14 **mysqli_close()** : **mysqli->close()**

Style procédural

bool **mysqli_close** (mysqli link)

Style orienté objet (méthode)

mysqli bool **close** ()

La fonction mysqli_close ferme la connexion spécifiée par le paramètre link .

8.81.15 **mysqli_commit()** : **mysqli->commit()**

Style procédural

bool **mysqli_commit** (mysqli link)

Style orienté objet (méthode)

mysqli bool **commit** ()

Valide la transaction courante pour la base de données spécifiée par le paramètre link .

8.81.16 **mysqli_connect_errno()** : Retourne le code d'erreur de la connexion MySQL

int **mysqli_connect_errno** ()

mysqli_connect_errno retourne le code d'erreur de la dernière tentative de connexion avec mysqli_connect . Si aucune erreur ne survient, la fonction retourne zéro.

Note

Les messages d'erreur du client MySQL sont disponibles dans `errmsg.h` , Les messages d'erreur du serveur MySQL sont disponibles dans `mysqld_error.h` . Dans la distribution source de MySQL, vous pouvez trouver la liste complète des messages d'erreurs et des codes d'erreurs dans le fichier `Docs/mysqld_error.txt` .

8.81.17 `mysqli_connect_error()` : Retourne le message d'erreur de connexion MySQL

string `mysqli_connect_error` ()

`mysqli_connect_error` retourne un message d'erreur, lié au dernier appel à `mysqli_connect`, qui correspond au numéro d'erreur retourné par `mysqli_connect_errno`. Si aucune erreur ne survient, cette fonction retourne la chaîne vide.

8.81.18 `mysqli_connect()` : `mysqli()`

Style procédural

`mysqli` **`mysqli_connect`** (*string* `host` , *string* `username` , *string* `passwd` , *string* `dbname` , *int* `port` , *string* `socket`)

Style orienté objet (méthode)

`mysqli` **`__construct`** *string* `host` *string* `username` *string* `passwd` *string* `dbname` *int* `port` *string* `socket`

La fonction `mysqli_connect` essaie d'ouvrir une connexion au serveur MySQL de l'hôte `host` qui peut être un nom d'hôte ou une adresse IP. Si vous passez la valeur NULL ou la chaîne "localhost" à ce paramètre, l'hôte local est sous-entendu. Lorsque c'est possible, les sockets seront utilisées au lieu du protocole TCP/IP. `mysqli_connect` retourne une ressource de connexion au serveur MySQL en cas de succès et, sinon, FALSE.

Les paramètres `username` et `password` spécifient l'utilisateur et le mot de passe avec lesquels on se connecte au serveur MySQL. Si le mot de passe n'est pas indiqué (la valeur NULL est passée), le serveur MySQL essaiera d'identifier l'utilisateur en étudiant que les enregistrements où les utilisateurs n'ont pas de mot de passe. Cela permet à un utilisateur de jouir de plusieurs permissions (selon que l'on fournit le mot de passe ou non).

Le paramètre `dbname`, si fourni, spécifiera la base de données par défaut à utiliser lors de l'exécution de requêtes.

Les paramètres `port` et `socket` sont utilisés en conjonction avec le paramètre `hostname` pour mieux contrôler la manière dont la connexion au serveur se fait. Le paramètre `port` spécifie le numéro de port à utiliser, tandis que le paramètre `socket` spécifie la socket ou le tunnel nommé qui doit être utilisé.

Note
Spécifier le paramètre <code>socket</code> ne déterminera pas explicitement le type de connexion qui sera utilisé lors de la connexion au serveur MySQL. Cela est déterminé par le paramètre <code>host</code> .

8.81.19 `mysqli_data_seek()` : `result->data_seek()`

Style procédural

bool **`mysqli_data_seek`** (`mysqli_result` `result` , *int* `offset`)

Style orienté objet (méthode)

mysqli_result bool **data_seek** (int offset)

La fonction mysqli_data_seek déplace le pointeur interne de résultat associé au jeu de résultat représenté par result , en le faisant pointer sur la ligne spécifiée par offset . Le paramètre offset doit être compris entre zéro et mysqli_num_rows - 1 (0.. mysqli_num_rows - 1).

Note

Cette fonction ne peut être utilisée qu'avec des résultats obtenus avec la fonction <u>mysqli_store_result</u> ou <u>mysqli_query</u> .

8.81.20 mysqli_debug() : Effectue des actions de débogage

bool **mysqli_debug** (string debug)

La fonction mysqli_debug est utilisée pour effectuer des actions de débogage en utilisant la bibliothèque de débogage Fred Fish. Le paramètre debug est une chaîne de caractères représentant l'opération de débogage à effectuer.

Note

Pour utiliser la fonction <u>mysqli_debug</u> , vous devez compiler la bibliothèque cliente MySQL avec le support du débogage.
--

8.81.21

mysqli_disable_reads_from_master() : mysqli->disable_reads_from_master()

Style procédural

bool **mysqli_disable_reads_from_master** (mysqli link)

Style orienté objet (méthode)

mysqli void **disable_reads_from_master** ()

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.81.22 mysqli_disable_rpl_parse() : Désactive le parseur RPL

bool **mysqli_disable_rpl_parse** (mysqli link)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.81.23 mysqli_dump_debug_info() : mysqli->dump_debug_info()

bool **mysqli_dump_debug_info** (mysqli link)

Cette fonction est à utiliser par un utilisateur possédant le privilège SUPER et est utilisée pour écrire quelques informations de débogage dans le log pour le serveur MySQL relatif à la connexion spécifiée par le paramètre link .

8.81.24 `mysqli_embedded_connect()` : Ouvre une connexion avec un serveur MySQL intégré

`mysqli` `mysqli_embedded_connect` (*string* *dbname*)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.81.25 `mysqli_enable_reads_from_master()` : Active la lecture depuis le maître

`bool` `mysqli_enable_reads_from_master` (`mysqli` *link*)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.81.26 `mysqli_enable_rpl_parse()` : Active le parseur RPL

`bool` `mysqli_enable_rpl_parse` (`mysqli` *link*)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.81.27 `mysqli_errno()` : `mysqli->errno()`

Style procédural

`int` `mysqli_errno` (`mysqli` *link*)

Style orienté objet (méthode)

`mysqli` `int` `errno`

La fonction `mysqli_errno` retournera le code erreur pour le dernier appel à une fonction MySQLi qui peut échouer ou réussir en respectant la connexion définie par la paramètre `link` . Si aucune erreur n'est survenue, cette fonction retournera zéro.

Note

Les numéros d'erreur client sont listés dans les en-têtes du fichier MySQL <code>errmsg.h</code> , les messages d'erreur du serveur sont listés dans le fichier <code>mysqld_error.h</code> . Dans les sources de MySQL, vous pouvez trouver une liste complète des messages d'erreur et des numéros d'erreur dans le fichier <code>Docs/mysqld_error.txt</code> .
--

8.81.28 `mysqli_error()` : Retourne une chaîne décrivant la dernière erreur

Style procédural

`string` `mysqli_error` (`mysqli` *link*)

Style orienté objet (méthode)

mysqli string error

La fonction mysqli_error est identique à la fonction mysqli_errno , sauf qu'au lieu de renvoyer un code erreur sous forme d'entier, cette fonction retournera une chaîne de caractères décrivant la dernière erreur survenue sur la connexion représentée par le paramètre link . Si aucune erreur n'est survenue, cette fonction retournera une chaîne vide.

8.81.29 mysqli_escape_string() : Alias de mysqli_real_escape_string

mysqli_escape_string est un alias de la fonction mysqli_real_escape_string .

8.81.30 mysqli_execute() : Alias de mysqli_stmt_execute

mysqli_execute est un alias de la fonction mysqli_stmt_execute . Pour une description détaillée, voir la description de la fonction mysqli_stmt_execute .

Note
<u>mysqli_execute</u> est fortement déconseillée et devrait être supprimée prochainement.

8.81.31 mysqli_fetch_array() : result->fetch_array()

Style procédural

mixed **mysqli_fetch_array** (mysqli_result result , int resulttype)

Style orienté objet (méthode)

mysqli_result mixed **fetch_array** (int resulttype)

Retourne un tableau qui correspond à la ligne lue ou FALSE s'il n'y a plus de lignes dans le jeu de résultats result .

mysqli_fetch_array est une version étendue de la fonction mysqli_fetch_row . En plus d'enregistrer les données sous forme d'un tableau à indices numériques, elle peut aussi les enregistrer dans un tableau associatif, en utilisant les noms des champs comme clés.

Note
Les noms des champs retournés par cette fonction sont sensibles à la casse .
Note
Cette fonction définit les champs NULL à la valeur PHP NULL .

Si plusieurs colonnes portent le même nom, la dernière colonne aura la priorité. Pour accéder aux autres colonnes du même nom, vous devez utiliser l'index numérique, ou faire un alias pour chaque colonne.

Le second argument optionnel resulttype est une constante indiquant quel type de tableau doit être renvoyé à partir de la ligne de données courante. Les valeurs possibles pour ce paramètre sont les constantes MYSQLI_ASSOC, MYSQLI_NUM, et MYSQLI_BOTH. MYSQLI_BOTH est la valeur par défaut de ce paramètre.

En utilisant la constante MYSQLI_ASSOC, cette fonction se comportera comme la fonction

mysqli_fetch_assoc , tandis que MYSQLI_NUM la fera agir comme la fonction mysqli_fetch_row . La constante MYSQLI_BOTH, créera elle un tableau qui sera à la fois associatif et indexé numériquement.

8.81.32 mysqli_fetch_assoc() : mysqli->fetch_assoc()

Style procédural

array **mysqli_fetch_assoc** (mysqli_result result)

Style orienté objet (méthode)

mysqli_result array **fetch_assoc** ()

Retourne un tableau associatif qui correspond à la ligne récupérée ou FALSE s'il n'y a plus de ligne.

La fonction mysqli_fetch_assoc est utilisée pour retourner un tableau associatif représentant la prochaine ligne dans le jeu de résultats représenté par le paramètre result , où chaque clé du tableau représente le nom d'une colonne du résultat.

Si deux ou plus colonnes dans le jeu de résultat ont le même nom, le tableau associatif retourné par la fonction mysqli_fetch_assoc ne contiendra que la valeur de la dernière colonne de ce nom. Si vous devez travailler avec des jeux de résultats ayant cette particularité, la fonction mysqli_fetch_row qui retourne un tableau indexé doit être utilisée à la place.

Note
Les noms des champs retournés par cette fonction sont sensibles à la casse .
Note
Cette fonction définit les champs NULL à la valeur PHP NULL .

8.81.33 mysqli_fetch_field_direct() : result->fetch_field_direct()

Style procédural

object **mysqli_fetch_field_direct** (mysqli_result result , int fieldnr)

Style orienté objet (méthode)

mysqli_result object **fetch_field_direct** (int fieldnr)

mysqli_fetch_field_direct retourne un objet qui contient les données métas d'un champ depuis le jeu de résultats result . Les valeurs de fieldnr doivent être dans l'intervalle 0 à nombre de champs - 1 .

8.81.34 mysqli_fetch_field() : result->fetch_field()

Style procédural

object **mysqli_fetch_field** (mysqli_result result)

Style orienté objet (méthode)

mysqli_result object **fetch_field** (void)

mysqli_fetch_field est utilisée pour retourner les attributs de la prochaine colonne dans le jeu de résultats représenté par le paramètre result en tant qu'objet. Appelez cette fonction de façon répétitive pour récupérer les informations de toutes les colonnes. Lors de son exécution, cette fonction retournera un objet contenant les attributs de la colonne courante ou FALSE s'il n'y a plus de colonnes dans le jeu de résultats.

8.81.35 mysqli_fetch_fields() : result->fetch_fields()

Style procédural

array **mysqli_fetch_fields** (mysqli_result result)

Style orienté objet (méthode)

mysqli_result array **fetch_fields** (void)

Cette fonction fonctionne comme mysqli_fetch_field à la différence que, au lieu de retourner un objet à la fois pour chaque champ, les colonnes sont retournées en tant que tableau d'objets.

8.81.36 mysqli_fetch_lengths() : result->lengths()

Style procédural

array **mysqli_fetch_lengths** (mysqli_result result)

Style orienté objet (méthode)

mysqli_result array lengths

La fonction mysqli_fetch_lengths retourne un tableau contenant la longueur de chaque colonne de la ligne courante du jeu de résultats représenté par le paramètre result . Un tableau indexé numériquement représentant la longueur de chaque colonne est retourné en cas de succès, FALSE en cas d'échec.

8.81.37 mysqli_fetch_object() : result->fetch_object()

Style procédural

mixed **mysqli_fetch_object** (mysqli_result result)

Style orienté objet (méthode)

mysqli_result mixed **fetch_object** ()

La fonction mysqli_fetch_object retourne la ligne courante du jeu de résultat sous forme d'objet dont les attributs représentent les noms des champs trouvés dans le jeu de résultats. S'il n'y a plus de lignes à lire, la fonction retourne FALSE .

8.81.38 `mysqli_fetch_row()` : `result->fetch_row()`

Style procédural

mixed **`mysqli_fetch_row`** (`mysqli_result` `result`)

Style orienté objet (méthode)

`mysqli_result` mixed **`fetch_row`** ()

Retourne un tableau correspondant à la ligne récupérée ou FALSE s'il n'y a plus de ligne.

`mysqli_fetch_row` récupère une ligne de données à partir du jeu de résultats représenté par `result` et le retourne en tant que tableau indexé, où chaque colonne est une case du tableau, celui-ci commençant à 0 (zéro). Chaque nouvel appel à `mysqli_fetch_row` retournera la prochaine ligne dans le jeu de résultats, ou FALSE s'il n'y a plus de lignes.

Note
Cette fonction définit les champs NULL à la valeur PHP NULL .

8.81.39 `mysqli_fetch()` : Alias de `mysqli_stmt_fetch`

`mysqli_fetch` est un alias de la fonction `mysqli_stmt_fetch` . Pour une description détaillée, voir la description de la fonction `mysqli_stmt_fetch` .

Note
<code>mysqli_fetch</code> est fortement déconseillée et devrait être supprimée prochainement.

8.81.40 `mysqli_field_count()` : `mysqli->field_count()`

Style procédural

int **`mysqli_field_count`** (`mysqli` `link`)

Style orienté objet (méthode)

`mysqli_result` int **`field_count`** ()

Retourne le nombre de colonnes pour la dernière requête sur la connexion spécifiée par le paramètre `link` . Cette fonction peut être utile lors de l'utilisation de `mysqli_store_result` pour déterminer si la requête aurait du retourner un résultat vide ou non, sans en connaître la nature.

8.81.41 `mysqli_field_seek()` : `result->field_seek()`

Style procédural

bool **`mysqli_field_seek`** (`mysqli_result` `result` , int `fieldnr`)

Style orienté objet (méthode)

`mysqli_result` bool **`field_seek`** (int `fieldnr`)

mysqli_field_seek place le curseur sur le champ spécifié par le numéro fieldnr . Le prochain appel à la fonction mysqli_fetch_field retournera la définition du champ de la colonne associée à cette position.

Note

Pour se déplacer au début d'une ligne, passez une position ayant pour valeur zéro.
--

8.81.42 mysqli_field_tell() : result->current_field()

Style procédural

```
int mysqli_field_tell ( mysqli_result result )
```

Style orienté objet (méthode)

```
mysqli_result int current_field
```

mysqli_field_tell retourne la position du champ dans un pointeur utilisée par le dernier appel à mysqli_fetch_field . Cette valeur peut être utilisée comme argument à la fonction mysqli_field_seek .

8.81.43 mysqli_free_result() : result->free()

Style procédural

```
void mysqli_free_result ( mysqli_result result )
```

Style orienté objet (toutes les méthodes sont équivalentes)

```
mysqli_result void free ( ) void close ( ) void free_result ( )
```

La fonction mysqli_free_result libère la mémoire associée au résultat représenté par le paramètre result , qui a été allouée par mysqli_query , mysqli_store_result ou mysqli_use_result .

Note

Vous devriez toujours libérer les résultats avec <u>mysqli_free_result</u> , lorsque votre objet de résultat ne vous est plus utile.
--

8.81.44 mysqli_get_client_info() : Retourne une chaîne contenant la version du client MySQL

```
string mysqli_get_client_info ( )
```

La fonction mysqli_get_client_info est utilisée pour retourner une chaîne de caractères représentant la version du client utilisé par l'extension MySQLi.

8.81.45 mysqli_get_client_version() : Lit les informations du client MySQL

```
int mysqli_get_client_version ( void )
```

[mysqli_get_client_version](#) retourne la version du client MySQL sous la forme d'un entier.

8.81.46 [mysqli_get_host_info\(\)](#) : [mysqli->get_host_info\(\)](#)

Style procédural

string [mysqli_get_host_info](#) (mysqli link)

Style orienté objet (méthode)

mysqli string host_info

La fonction [mysqli_get_host_info](#) retourne une chaîne de caractères décrivant la connexion représentée par le paramètre link (incluant le nom d'hôte du serveur).

8.81.47 [mysqli_get_metadata\(\)](#) : Alias de [mysqli_stmt_result_metadata](#)

[mysqli_get_metadata](#) est un [alias](#) de la fonction [mysqli_stmt_result_metadata](#) . Pour une description détaillée, reportez-vous à la documentation de [mysqli_stmt_result_metadata](#) .

Note

mysqli_get_metadata est obsolète, et sera bientôt supprimée.
--

8.81.48 [mysqli_get_proto_info\(\)](#) : [mysqli->protocol_version\(\)](#)

Style procédural

int [mysqli_get_proto_info](#) (mysqli link)

Style orienté objet (méthode)

mysqli string protocol_version

[mysqli_get_proto_info](#) retourne un entier représentant la version du protocole MySQL utilisé par la connexion représentée par le paramètre link .

8.81.49 [mysqli_get_server_info\(\)](#) : [mysqli->server_info\(\)](#)

Style procédural

string [mysqli_get_server_info](#) (mysqli link)

Style orienté objet (méthode)

mysqli string server_info

Retourne une chaîne de caractères représentant la version du serveur MySQL auquel l'extension MySQLi est connectée (représenté par le paramètre link).

8.81.50 `mysqli_get_server_version()` : Retourne un entier représentant la version du serveur MySQL

Style procédural

int `mysqli_get_server_version` (`mysqli link`)

Style orienté objet (méthode)

`mysqli` int `server_version`

La fonction `mysqli_get_server_version` retourne un entier représentant la version du serveur sur lequel on est connecté (représenté par le paramètre `link`).

Le format de ce nombre est `main_version * 10000 + minor_version * 100 + sub_version` (i.e. la version 4.1.0 retournera 40100).

8.81.51 `mysqli_info()` : `mysqli->info()`

Style procédural

string `mysqli_info` (`mysqli link`)

Style orienté objet (méthode)

`mysqli` string `info`

La fonction `mysqli_info` retourne une chaîne fournissant des informations à propos de la dernière requête exécutée. La nature de cette chaîne est fournie ci-dessous :

Type de requête	Exemple de retour
INSERT INTO...SELECT...	Records: 100 Duplicates: 0 Warnings: 0
INSERT INTO...VALUES (...),(...),(...)	Records: 3 Duplicates: 0 Warnings: 0
LOAD DATA INFILE ...	Records: 1 Deleted: 0 Skipped: 0 Warnings: 0
ALTER TABLE ...	Records: 3 Duplicates: 0 Warnings: 0
UPDATE ...	Rows matched: 40 Changed: 40 Warnings: 0
Note	
Les requêtes qui ne font pas partie de la liste précédente ne sont pas supportées. Dans cette situation, <code>mysqli_info</code> retournera FALSE	

8.81.52 `mysqli_init()` : Initialise MySQLi et retourne une ressource à utiliser avec `mysqli_real_connect()`

`mysqli` `mysqli_init` (void)

`mysqli_init` alloue ou initialise un objet MySQL utilisable pour les fonctions `mysqli_options` et `mysqli_real_connect` .

Note

Tous les appels suivants à n'importe quelle fonction MySQLi (exceptée <code>mysqli_options</code>) échoueront en attendant que la fonction <code>mysqli_real_connect</code> soit appelée.
--

8.81.53 `mysqli_insert_id()` : `mysqli->insert_id()`

Style procédural

int `mysqli_insert_id` (`mysqli link`)

Style orienté objet (méthode)

`mysqli` int `insert_id`

La fonction `mysqli_insert_id` retourne l'identifiant généré par une requête sur une table avec une colonne possédant l'attribut `AUTO_INCREMENT`. Si la dernière requête n'était ni un `INSERT`, ni un `UPDATE` ou que la table modifiée ne possède pas de colonne avec l'attribut `AUTO_INCREMENT`, cette fonction retournera zéro.

Note

Exécuter une commande <code>INSERT</code> ou <code>UPDATE</code> utilisant la fonction <code>LAST_INSERT_ID()</code> modifiera aussi la valeur retournée par la fonction <code>mysqli_insert_id</code> .
--

8.81.54 `mysqli_kill()` : `mysqli->kill()`

Style procédural

bool `mysqli_kill` (`mysqli link` , int `processid`)

Style orienté objet (méthode)

`mysqli` bool `kill` (int `processid`)

`mysqli_kill` est utilisée pour demander au serveur de terminer un thread MySQL spécifié par le paramètre `processid`. Cette valeur doit être obtenue en appelant la fonction `mysqli_thread_id`.

Note

Pour arrêter une requête en cours d'exécution, utilisez la commande SQL <code>KILL QUERY processid</code> .

8.81.55 `mysqli_master_query()` : Force l'exécution d'une requête sur le maître dans une configuration maître/esclave (master/slave)

bool `mysqli_master_query` (`mysqli link` , string `query`)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.81.56 `mysqli_more_results()` : `mysqli->more_results()`

bool `mysqli_more_results` (`mysqli link`)

`mysqli_more_results` indique si un ou plusieurs jeux de résultats sont disponibles, générés par un appel antérieur à `mysqli_multi_query`.

8.81.52 `mysqli_init()` : Initialise MySQLi et retourne une ressource à utiliser avec `mysqli_real_connect()`

8.81.57 `mysqli_multi_query()` : `mysqli->multi_query()`

Style procédural

bool `mysqli_multi_query` (`mysqli link` , `string query`)

Style orienté objet (méthode)

`mysqli` bool `multi_query` (`string query`)

`mysqli_multi_query` exécute une ou plusieurs requêtes, rassemblées dans le paramètre `query` par des point-virgules.

Pour lire les résultats de la première requête, vous pouvez utiliser les fonctions `mysqli_use_result` et `mysqli_store_result` . Tous les autres résultats de requêtes peuvent être atteints avec `mysqli_more_results` et `mysqli_next_result` .

8.81.58 `mysqli_next_result()` : `mysqli->next_result()`

bool `mysqli_next_result` (`mysqli link`)

`mysqli_next_result` prépare le prochain jeu de résultat, initialisé par un appel antérieur à `mysqli_multi_query` , et qui peut être lu par `mysqli_store_result` ou `mysqli_use_result` .

8.81.59 `mysqli_num_fields()` : `result->field_count()`

Style procédural

int `mysqli_num_fields` (`mysqli_result result`)

Style orienté objet (méthode)

`mysqli_result` int `field_count`

`mysqli_num_fields` retourne le nombre de champs dans le jeu de résultats spécifié `result` .

8.81.60 `mysqli_num_rows()` : Retourne le nombre de lignes dans un résultat

Style procédural

int `mysqli_num_rows` (`mysqli result`)

Style orienté objet (méthode)

`mysqli` int `num_rows`

`mysqli_num_rows` retourne le nombre de lignes dans un jeu de résultats.

L'utilisation de `mysqli_num_rows` dépend de l'utilisation de jeux de résultats bufférisés ou non. Dans le cas où vous utilisez des jeux de résultats non bufférisés, `mysqli_num_rows` ne retournera pas le nombre correct de lignes tant que toutes les lignes du jeu de résultats ne sont pas retournées.

8.81.61 `mysqli_options()` : `mysqli->options()`

Style procédural

bool **`mysqli_options`** (`mysqli link` , int `option` , mixed `value`)

Style orienté objet (méthode)

mysqli bool **`options`** (int `option` , mixed `value`)

`mysqli_options` peut être utilisée pour définir des options de connexion et ainsi affecter le comportement de la connexion courante.

Cette fonction peut être appelée plusieurs fois pour définir plusieurs options.

`mysqli_options` doit être appelée après `mysqli_init` et avant `mysqli_real_connect` .

Le paramètre `option` est l'option que vous voulez définir, et `value` est sa valeur. Le paramètre `option` peut être une des valeurs suivantes :

Nom	Description
<code>MYSQLI_OPT_CONNECT_TIMEOUT</code>	timeout de la connexion en secondes
<code>MYSQLI_OPT_LOCAL_INFILE</code>	active/désactive l'utilisation de LOAD LOCAL INFILE
<code>MYSQLI_INIT_COMMAND</code>	commande à exécuter après la connexion au serveur MySQL
<code>MYSQLI_READ_DEFAULT_FILE</code>	Lit les options depuis le nom de l'option plutôt que du fichier <code>my.cnf</code>
<code>MYSQLI_READ_DEFAULT_GROUP</code>	Lit les options du groupe depuis <code>my.cnf</code> ou depuis le fichier spécifié avec <code>MYSQLI_READ_DEFAULT_FILE</code> .

8.81.62 `mysqli_param_count()` : Alias de `mysqli_stmt_param_count`

`mysqli_param_count` est un alias de la fonction `mysqli_stmt_param_count` . Pour une description détaillée, voir la description de la fonction `mysqli_stmt_param_count` .

Note
<code>mysqli_param_count</code> est fortement déconseillée et devrait être supprimée prochainement.

8.81.63 `mysqli_ping()` : `mysqli->ping()`

Style procédural

bool **`mysqli_ping`** (`mysqli link`)

Style orienté objet (méthode)

mysqli bool **ping** (void)

Ping la connexion pour s'assurer que le serveur est bien en fonctionnement. S'il n'est pas en fonctionnement, et que l'option global `mysqli.reconnect` est activée, une connexion automatique sera tentée.

Cette fonction peut être utilisée par les clients qui font de lourds traitements, pour vérifier si le serveur a fermé la connexion et, ainsi, tenter une reconnexion si nécessaire.

8.81.64 `mysqli_prepare()` : `mysqli->prepare()`

Style procédural

mysqli_stmt **mysqli_prepare** (mysqli link , string query)

Style orienté objet (méthode)

mysqli mysqli_stmt **prepare** (string query)

`mysqli_prepare` prépare la requête SQL `query` et retourne une ressource à utiliser pour les futures opérations sur le traitement. La requête doit être une requête SQL unique.

Note

Vous ne devez pas ajouter de point virgule ou de `\g` dans la requête.

Le paramètre `query` peut inclure un ou plusieurs paramètres de marques dans la requête SQL avec le caractère "point d'interrogation" (?) à la position appropriée.

Note

Les marques sont autorisées uniquement dans certains endroits des requêtes SQL. Par exemple, elles le sont dans la liste `VALUES()` d'une requête `INSERT` (pour spécifier les valeurs des colonnes pour une ligne), ou dans une comparaison d'une clause `WHERE` pour spécifier une valeur de comparaison.

Cependant, elles ne sont pas autorisées pour les identifiants (comme les noms de tables ou de colonnes), dans une liste de sélection où les noms des colonnes doivent être retournés par une requête `SELECT`, ou pour spécifier un opérateur tel que le signe égal (=). La dernière restriction est nécessaire car il est impossible de déterminer le type de paramètre. Il n'est pas non plus autorisé de comparer les marqueurs avec `NULL` par `? IS NULL`. En général, les paramètres ne sont autorisés que dans les requêtes DML (Data Manipulation Language) et non dans les requêtes DDL (Data Definition Language).

Les paramètres de marques doivent être liés à des variables utilisées dans les fonctions `mysqli_stmt_bind_param` et/ou `mysqli_stmt_bind_result` avant d'exécuter la requête ou de récupérer les lignes.

8.81.65 `mysqli_query()` : `mysqli->query()`

Style procédural

mixed **mysqli_query** (mysqli link , string query , int resultmode)

Style orienté objet (méthode)

8.81.63 `mysqli_ping()` : `mysqli->ping()`

mysqli mixed **query** (string query , int resultmode)

La fonction mysqli_query est utilisée pour simplifier l'acte d'exécuter une requête sur la connexion à la base de données représentée par le paramètre link .

En termes de fonctionnalités, utiliser cette fonction revient à appeler mysqli_real_query suivie de mysqli_use_result ou mysqli_store_result où query est la requête elle-même et resultmode la constante MYSQLI_USE_RESULT ou MYSQLI_STORE_RESULT selon le comportement voulu. Par défaut, si resultmode n'est pas renseigné, c'est la constante MYSQLI_STORE_RESULT qui est utilisée.

Si vous exécutez mysqli_query avec le paramètre resultmode valant MYSQLI_USE_RESULT , tous les appels suivants retourneront l'erreur Commands out of sync à moins que vous appeliez mysqli_free_result .

8.81.66 mysqli_real_connect() : mysqli->real_connect()

Style procédural

bool **mysqli_real_connect** (mysqli link , string hostname , string username , string passwd , string dbname , int port , string socket , int flags)

Style orienté objet (méthode)

mysqli bool **real_connect** (string hostname , string username , string passwd , string dbname , int port , string socket , int flags)

mysqli_real_connect tente d'établir une connexion avec un serveur MySQL d'un hôte.

Cette fonction diffère de mysqli_connect en ces points :

- mysqli_real_connect a besoin d'un objet valide issu de la fonction mysqli_init
- Avec la fonction mysqli_options , vous pouvez définir différentes options pour la connexion.
- Avec les paramètres flags , vous pouvez définir différents options de la connexion :

Nom	Description
MYSQLI_CLIENT_COMPRESS	Utilise un protocole de compression
MYSQLI_CLIENT_FOUND_ROWS	Retourne le nombre de lignes sélectionnées, et non pas le nombre de lignes affectées
MYSQLI_CLIENT_IGNORE_SPACE	Accepte les espaces après les noms de fonctions. Fait que tous les noms de fonctions sont des mots réservés.
MYSQLI_CLIENT_INTERACTIVE	Permet interactive_timeout secondes (au lieu de wait_timeout secondes) d'inactivité avant de fermer la connexion
MYSQLI_CLIENT_SSL	Utilise SSL (chiffrage)
Note	Pour des raisons de sécurité, le flag MULTI_STATEMENT n'est pas supporté en PHP. Si vous voulez exécuter plusieurs requêtes, utilisez la fonction <u>mysqli_multi_query</u> .

8.81.67 mysqli_real_escape_string() : mysqli->real_escape_string()

Style procédural

string **mysqli_real_escape_string** (mysqli link , string escapestr)

Style orienté objet (méthode)

mysqli string **real_escape_string** (string escapestr)

mysqli_real_escape_string est utilisée pour créer une chaîne SQL valide qui pourra être utilisée dans une requête SQL. La chaîne de caractères escapestr est encodée en une chaîne SQL échappée, en tenant compte du jeu de caractères courant de la connexion.

Les caractères encodés sont NUL (ASCII 0), \n, \r, \, ', ", et Control-Z .

8.81.68 mysqli_real_query() : mysqli->real_query()

Style procédural

bool **mysqli_real_query** (mysqli link , string query)

Style orienté objet (méthode)

mysqli bool **real_query** (string query)

La fonction mysqli_real_query est utilisée pour exécuter une seule requête sur la connexion à la base de données représentée par le paramètre link dont le résultat peut être récupéré ou stocké en utilisant les fonctions mysqli_store_result ou mysqli_use_result .

Note
Pour déterminer si une requête donnée aurait du retourner un jeu de résultats ou non, voyez la fonction <u>mysqli_field_count</u> .

8.81.69 mysqli_report() : Active ou désactive les fonctions de rapport internebool **mysqli_report** (int flags)

mysqli_report est une fonction puissante, pour améliorer vos requêtes et votre code durant les phases de développement et de tests. En fonction de l'argument flags , il rapporte des erreurs liées aux appels de fonctions mysqli ou aux requêtes qui n'utilisent pas d'index (ou n'utilisent pas de bon index).

Nom	Description
MYSQLI_REPORT_OFF	Désactive le rapport
MYSQLI_REPORT_ERROR	Rapporte les erreurs des fonctions mysqli
MYSQLI_REPORT_INDEX	Rapporte si un mauvais index ou pas d'index a été utilisé
MYSQLI_REPORT_ALL	Active toutes les options

8.81.70 `mysqli_rollback()` : `mysqli->rollback()`

Style procédural

bool `mysqli_rollback` (`mysqli link`)

Style orienté objet (méthode)

`mysqli` bool `rollback` (void)

Annule la transaction courante pour la base de données spécifiée par le paramètre `link` .

8.81.71 `mysqli_rpl_parse_enabled()` : Vérifie si le parseur RPL est activé

int `mysqli_rpl_parse_enabled` (`mysqli link`)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.81.72 `mysqli_rpl_probe()` : Sonde le RPL

bool `mysqli_rpl_probe` (`mysqli link`)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.81.73 `mysqli_rpl_query_type()` : `mysqli->rpl_query_type()`

Style procédural

int `mysqli_rpl_query_type` (`mysqli link` , string `query`)

Style orienté objet (méthode)

`mysqli` int `rpl_query_type` (string `query`)

Retourne `MYSQLI_RPL_MASTER` , `MYSQLI_RPL_SLAVE` ou `MYSQLI_RPL_ADMIN` dépendamment du type de requête. `INSERT` , `UPDATE` et similaire sont des requêtes **maîtres** , `SELECT` est **esclave** , et `FLUSH` , `REPAIR` et similaire sont des requêtes d' **administration** .

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.81.74 `mysqli_select_db()` : `mysqli->select_db()`

bool `mysqli_select_db` (`mysqli link` , string `dbname`)

La fonction `mysqli_select_db` sélectionne la base de données par défaut (spécifiée par le paramètre `dbname`) pour être utilisée lors de l'exécution de requêtes sur la connexion représentée par le paramètre `link` .

Note

Cette fonction ne doit être utilisée que pour changer la base de données par défaut pour la connexion courante. Vous pouvez sélectionner la base de données par défaut avec le 4ème paramètre de la fonction [mysqli_connect](#) .

8.81.75 [mysqli_send_long_data\(\)](#) : Alias de [mysqli_stmt_send_long_data](#)

[mysqli_send_long_data](#) est un [alias](#) de la fonction [mysqli_stmt_send_long_data](#) . Pour une description détaillée, voir la description de la fonction [mysqli_stmt_send_long_data](#) .

Note

[mysqli_send_long_data](#) est fortement déconseillée et devrait être supprimée prochainement.

8.81.76 [mysqli_send_query\(\)](#) : [mysqli->send_query\(\)](#)

Style procédural

bool [mysqli_send_query](#) (mysqli link , string query)

Style orienté objet (méthode)

mysqli bool [send_query](#) (string query)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.81.77 [mysqli_server_end\(\)](#) : Eteint le serveur embarqué

void [mysqli_server_end](#) (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.81.78 [mysqli_server_init\(\)](#) : Initialise le serveur MySQL intégré

bool [mysqli_server_init](#) (array server , array groups)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.81.79 [mysqli_set_charset\(\)](#) : [mysqli->set_charset\(\)](#)

bool [mysqli_set_charset](#) (mysqli link , string charset)

La fonction [mysqli_set_charset](#) définit le jeu de caractères par défaut (spécifié par le paramètre charset) à utiliser lors de l'envoi de données depuis et vers le serveur de base de données représenté par le paramètre link .

Note

Pour utiliser cette fonction sur les systèmes Windows, vous devez utiliser la bibliothèque client MySQL version 4.1.11 ou suivante (pour MySQL 5.0, vous avez besoin de la version 5.0.6 ou suivante).

8.81.80 `mysqli_set_opt()` : Alias de `mysqli_options`

`mysqli_set_opt` est un alias de la fonction `mysqli_options` .

8.81.81 `mysqli_sqlstate()` : `mysqli->sqlstate()`

Style procédural

string **`mysqli_sqlstate`** (mysqli link)

Style orienté objet (méthode)

mysqli string sqlstate

`mysqli_sqlstate` retourne une chaîne contenant le code d'erreur SQLSTATE de la dernière erreur. Le code d'erreur '00000' signifie : "pas d'erreur". Les valeurs sont spécifiées par les normes ANSI SQL et ODBC. Pour une liste des valeurs possibles, voyez : <http://dev.mysql.com/doc/mysql/en/error-handling.html> .

Note

Notez que toutes les erreurs de MySQL n'ont pas encore de correspondance avec les erreurs SQLSTATE. La valeur HY000 (erreur générale) est utilisée pour les erreurs sans correspondance.

8.81.82 `mysqli_ssl_set()` : `mysqli->ssl_set()`

Style procédural

bool **`mysqli_ssl_set`** (mysqli link , string key , string cert , string ca , string capath , string cipher)

Style orienté objet (méthode)

mysqli bool **`ssl_set`** (string key , string cert , string ca , string capath , string cipher)

`mysqli_ssl_set` est utilisée pour établir une connexion sécurisée avec SSL. Elle doit être appelée avant `mysqli_real_connect` . Cette fonction ne fait rien du tout si le support OpenSSL n'est pas activé.

key est le chemin vers le fichier des clés. cert est le chemin vers le fichier des certificats. ca est le chemin vers le fichier des certificats d'autorités. capath est le chemin vers le dossier qui contient les certificats SSL CA dans le format pem. cipher est la liste des chiffres autorisés à être utilisés pour le chiffrement SSL. Tous les paramètres SSL non utilisés peuvent avoir comme valeur NULL .

8.81.83 `mysqli_stat()` : `mysqli->stat()`

Style procédural

string **mysqli_stat** (mysqli link)

Style orienté objet (méthode)

mysqli string **stat** ()

mysqli_stat retourne une chaîne de caractères contenant des informations similaires à la commande 'mysqladmin status'. Cela inclut l'uptime en secondes et le nombre de threads courant, le nombre de commandes, les tables rechargées et ouvertes.

8.81.84 mysqli_stmt_affected_rows() : mysqli_stmt->affected_rows()

Style procédural

int **mysqli_stmt_affected_rows** (mysqli_stmt stmt)

Style orienté objet (méthode)

mysqli_stmt int **affected_rows**

mysqli_stmt_affected_rows retourne le nombre de lignes affectées par une requête INSERT, UPDATE ou DELETE. Si la dernière requête n'est pas valide, cette fonction retournera -1.

La fonction mysqli_stmt_affected_rows ne fonctionne que sur les requêtes qui modifient la table. Si vous désirez récupérer le nombre de lignes retournées par une requête SELECT, utilisez plutôt la requête mysqli_stmt_num_rows.

8.81.85 mysqli_stmt_bind_param() : stmt->bind_param()

Style procédural

bool **mysqli_stmt_bind_param** (mysqli_stmt stmt , string types , mixed var1 , *mixed* ...)

Style orienté objet (méthode)

mysqli_stmt bool **bind_param** (string types , mixed var1 , *mixed* ...)

mysqli_stmt_bind_param sert à lier des variables à une requête MySQL préparée par mysqli_prepare. La chaîne types contient un ou plusieurs caractères qui spécifient le type de la variable à lier :

Caractère	Description
i	Type entier
d	Type double (nombre à virgule flottante)
s	Type chaîne
b	Type BLOB, qui sera envoyé par paquets
Note	
Si la taille des données dépasse la taille maximal d'un paquet, (<code>max_allowed_packet</code>), vous devez spécifier le caractère b dans le paramètre types et utiliser la fonction <u>mysqli_stmt_send_long_data</u> pour envoyer le message par paquets.	

Le nombre de variables et la taille de la chaîne types doivent correspondre au nombre de variables de la requête.

8.81.86 `mysqli_stmt_bind_result()` : `stmt->bind_result()`

Style procédural

bool **mysqli_stmt_bind_result** (mysqli_stmt stmt , mixed var1 , *mixed* ...)

Style orienté objet (méthode)

mysqli_stmt bool **bind_result** (mixed var1 , *mixed* ...)

`mysqli_stmt_bind_result` est utilisée pour associer des colonnes d'un résultat à des variables. Lorsque `mysqli_stmt_fetch` est appelée pour lire des valeurs, le protocole MySQL place les données dans les variables spécifiées dans le paramètre var1,

Note

Notez que toutes les colonnes doivent être liées avant d'appeler `mysqli_stmt_fetch` . En fonction du type de valeur de la colonne, le type de variable PHP peut être automatiquement modifié.

Une colonne peut être associée ou réassociée à tout moment, même après une lecture partielle du résultat. La nouvelle association prend effet au prochain appel de `mysqli_stmt_fetch` .

8.81.87 `mysqli_stmt_close()` : `mysqli_stmt->close()`

Style procédural

bool **mysqli_stmt_close** (mysqli_stmt stmt)

Style orienté objet (méthode)

mysqli_stmt bool **close** ()

`mysqli_stmt_close` ferme une requête préparée. `mysqli_stmt_close` désaloue le pointeur utilisé par stmt . Si la requête est en attente ou bien que les résultats ne sont pas encore lus, cette fonction les annulera et, donc, la prochaine requête pourra être exécutée.

8.81.88 `mysqli_stmt_data_seek()` : `stmt->data_seek()`

Style procédural

void **mysqli_stmt_data_seek** (mysqli_stmt statement , int offset)

Style orienté objet (méthode)

mysqli_stmt void **data_seek** (int offset)

`mysqli_stmt_data_seek` déplace le pointeur du résultat statement de offset lignes. offset doit prendre une valeur entre zéro et le nombre total de ligne moins 1 (0.. `mysqli_stmt_num_rows` - 1).

8.81.89 mysqli_stmt_errno() : mysqli_stmt->errno()

Style procédural

int **mysqli_stmt_errno** (mysqli_stmt stmt)

Style orienté objet (méthode)

mysqli_stmt int **errno**

Pour le traitement identifié par `stmt` , `mysqli_stmt_errno` retourne le code erreur pour la dernière requête appelée dans le traitement qui a réussi ou échoué.

Note

La liste des codes erreur client est disponible dans l'en-tête du fichier MySQL `errmsg.h` ; la liste des codes erreur serveur est disponible dans le fichier MySQL `mysqld_error.h` . Dans les sources de MySQL, vous pouvez trouver une liste complète des messages d'erreur ainsi que des codes erreur dans le fichier `Docs/mysqld_error.txt` .

8.81.90 mysqli_stmt_error() : mysqli_stmt->error()

Style procédural

string **mysqli_stmt_error** (mysqli_stmt stmt)

Style orienté objet (méthode)

mysqli_stmt string **error**

Pour le traitement spécifié par `stmt` , `mysqli_stmt_error` retourne une chaîne de caractères représentant le message d'erreur le plus récent appelé par une fonction de traitement, qu'elle ait réussi ou échoué.

8.81.91 mysqli_stmt_execute() : stmt->execute()

Style procédural

bool **mysqli_stmt_execute** (mysqli_stmt stmt)

Style orienté objet (méthode)

mysqli_stmt bool **execute** ()

`mysqli_stmt_execute` exécute une requête qui a été préalablement préparée en utilisant la fonction `mysqli_prepare` , grâce à la ressource `stmt` . Lors de l'exécution, toutes les variables de la requête seront remplacées par les données appropriées.

Si la requête est UPDATE, DELETE ou INSERT, le nombre total de lignes affectées est disponible via la fonction `mysqli_stmt_affected_rows` . De la même façon, si la fonction génère un résultat, il sera disponible via la fonction `mysqli_stmt_fetch` .

Note

Lors de l'utilisation de [mysqli_stmt_execute](#) , [mysqli_stmt_fetch](#) doit être utilisée pour lire les données avant de lancer une autre requête.

8.81.92 [mysqli_stmt_fetch\(\)](#) : [stmt->fetch\(\)](#)

Style procédural

bool **mysqli_stmt_fetch** (mysqli_stmt stmt)

Style orienté objet (méthode)

mysqli_stmt bool **fetch** ()

[mysqli_stmt_fetch](#) retourne le résultat d'une requête préparée dans une variable, liée par [mysqli_stmt_bind_result](#) .

Note

Notez que toutes les colonnes doivent être liées par l'application avant d'appeler [mysqli_stmt_fetch](#) .

8.81.93 [mysqli_stmt_free_result\(\)](#) : [stmt->free_result\(\)](#)

Style procédural

void **mysqli_stmt_free_result** (mysqli_stmt stmt)

Style orienté objet (méthode)

mysqli_stmt void **free_result** ()

[mysqli_stmt_free_result](#) libère le résultat stmt de la mémoire. stmt a été obtenu de la fonction [mysqli_stmt_store_result](#) .

8.81.94 [mysqli_stmt_init\(\)](#) : [mysqli->stmt_init\(\)](#)

Style procédural

mysqli_stmt **mysqli_stmt_init** (mysqli link)

Style orienté objet (méthode)

mysqli_stmt **stmt_init** (void)

[mysqli_stmt_init](#) alloue et initialise un objet de commande, à utiliser avec [mysqli_stmt_prepare](#) .

Note

Tous les appels ultérieurs aux fonctions [mysqli_stmt_*](#) échoueront, si [mysqli_stmt_prepare](#) est appelée.

8.81.95 `mysqli_stmt_num_rows()` : `stmt->num_rows()`

Style procédural

```
int mysqli_stmt_num_rows ( mysqli_stmt stmt )
```

Style orienté objet (méthode)

```
mysqli_stmt int num_rows
```

mysqli_stmt_num_rows retourne le nombre de lignes dans le résultat `stmt` . L'utilisation de la fonction `mysqli_stmt_num_rows` dépend de l'utilisation ou non de la fonction `mysqli_stmt_store_result` pour stocker le résultat dans la ressource de commande.

Si vous utilisez `mysqli_stmt_store_result` , `mysqli_stmt_num_rows` peut être appelée immédiatement.

8.81.96 `mysqli_stmt_param_count()` : `stmt->param_count()`

Style procédural

```
int mysqli_stmt_param_count ( mysqli_stmt stmt )
```

Style orienté objet (méthode)

```
mysqli_stmt int param_count
```

`mysqli_stmt_param_count` retourne le nombre de variables attendues dans la requête préparée `stmt` .

8.81.97 `mysqli_stmt_prepare()` : `stmt->prepare()`

Style procédural

```
bool mysqli_stmt_prepare ( mysqli_stmt stmt , string query )
```

Style orienté objet (méthode)

```
mysqli_stmt mixed prepare ( string query )
```

`mysqli_stmt_prepare` prépare la requête SQL `query` , pour la session de travail `stmt` . `stmt` doit avoir été alloué par `mysqli_stmt_init` . La requête doit être une seule requête SQL.

Note

Il ne faut pas ajouter de point-virgule ou de <code>\g</code> à la fin de la requête.

Le paramètre `query` peut inclure une ou plusieurs variables SQL, en utilisant des points d'interrogation (?) aux bons endroits.

Note

Les variables SQL ne sont possibles que dans certaines clauses de la requête SQL. Par exemple, elles peuvent être placées dans des clause <code>VALUES()</code> d'une requête <code>INSERT</code> (pour spécifier une

valeur à insérer), ou dans une clause de condition WHERE.

Cependant, elles ne sont pas autorisées pour les identifiants (de tables ou de colonnes), dans les listes de colonnes d'un SELECT, ou pour spécifier des opérateurs comme = . Cette dernière restriction est liée au fait qu'il est impossible de déterminer le type. En général, les variables SQL ne sont valides que dans les commandes de manipulation de données (Data Manipulation Language (DML)), et non dans les structures du langage SQL (Data Definition Language (DDL)).

Les variables SQL doivent être associées à une variable PHP à l'aide de la fonction `mysqli_stmt_bind_param` et/ou `mysqli_stmt_bind_result` , avant d'exécuter la requête.

8.81.98 `mysqli_stmt_reset()` : `stmt->reset()`

Style procédural

bool `mysqli_stmt_reset` (`mysqli_stmt` `stmt`)

Style orienté objet (méthode)

`mysqli_stmt` bool `reset` ()

`mysqli_stmt_reset` annule une requête préparée sur le client et sur le serveur après avoir été préparée. Actuellement, cette fonction est essentiellement utilisée pour annuler les données envoyées avec la fonction `mysqli_stmt_send_long_data` .

Note

Pour préparer de nouveau une requête, utilisez la fonction `mysqli_stmt_prepare` .

8.81.99 `mysqli_stmt_result_metadata()` : Retourne les méta-données de préparation de requête MySQL

Style procédural

`mysqli_result` `mysqli_stmt_result_metadata` (`mysqli_stmt` `stmt`)

Style orienté objet (méthode)

`mysqli_stmt` `mysqli_result` `result_metadata` ()

Si une commande a été préparée par `mysqli_prepare` , et qu'elle produira un résultat, `mysqli_stmt_result_metadata` retourne l'objet de résultat qui sera utilisé pour lire les méta-données, comme le nombre de champs et les informations de colonnes.

Note

Ce résultat peut être passé comme argument à toutes les fonctions qui demandent un champ, pour y lire les méta-données :

- `mysqli_num_fields`
- `mysqli_fetch_field`
- `mysqli_fetch_field_direct`
- `mysqli_fetch_fields`
- `mysqli_field_count`

- [mysqli_field_seek](#)
- [mysqli_field_tell](#)
- [mysqli_free_result](#)

Il est recommandé de libérer la ressource de résultat lorsque vous avez terminé de l'utiliser, en la passant à la fonction [mysqli_free_result](#)

Note

Le jeu de résultat retourné par [mysqli_stmt_result_metadata](#) ne contient que des méta-données. Il ne contient aucune ligne de résultat. Ces lignes sont obtenues en utilisant la fonction [mysqli_stmt_fetch](#).

8.81.100 [mysqli_stmt_send_long_data\(\)](#) : [stmt->send_long_data\(\)](#)

Style procédural

bool [mysqli_stmt_send_long_data](#) (mysqli_stmt stmt , int param_nr , string data)

Style orienté objet (méthode)

mysqli_stmt bool [send_long_data](#) (int param_nr , string data)

[mysqli_stmt_send_long_data](#) envoie les données au serveur par paquets, si la taille des données excède la limite de `max_allowed_packet`. Cette fonction peut être appelée plusieurs fois pour envoyer les données textes ou binaires de champs comme les BLOB ou TEXT.

param_nr indique quel paramètre doit être associé avec quelles données. Les paramètres sont numérotés à partir de 0. data est une chaîne de caractères, contenant les données à envoyer.

8.81.101 [mysqli_stmt_sqlstate\(\)](#) : Retourne le code SQLSTATE de la dernière opération MySQL

string [mysqli_stmt_sqlstate](#) (mysqli_stmt stmt)

[mysqli_stmt_sqlstate](#) retourne une chaîne contenant le code d'erreur SQLSTATE de la commande la plus récente qui a été préparée. Le code d'erreur est constitué de 5 caractères. '00000' Les valeurs des codes d'erreurs sont spécifiées par les normes ANSI SQL et ODBC. Pour la liste complète des valeurs, voyez le fichier <http://dev.mysql.com/doc/mysql/en/error-handling.html>.

Note

Notez que toutes les erreurs MySQL n'ont pas encore de correspondance en SQLSTATE. La valeur HY000 (erreur générale) est utilisée pour les erreurs sans correspondance.

8.81.102 [mysqli_stmt_store_result\(\)](#) : [mysqli_stmt->store_result\(\)](#)

Style procédural

bool [mysqli_stmt_store_result](#) (mysqli_stmt stmt)

Style orienté objet (méthode)

mysqli_stmt bool [store_result](#) ()

Vous devez appeler `mysqli_stmt_store_result` pour toutes les requêtes qui produisent un jeu de résultats (SELECT, SHOW, DESCRIBE, EXPLAIN), et uniquement si vous voulez stocker le jeu de résultats complet par le client, et donc, les séquences `mysqli_stmt_fetch` pourront retourner ces données stockées.

Note

Il n'est pas nécessaire d'appeler `mysqli_stmt_store_result` pour d'autres types de requête, mais si vous le faites, ce n'est pas grave et ne causera aucune perte notable de performance dans tous les cas. Vous pouvez détecter dans tous les cas si votre requête va produire un jeu de résultats en regardant si la fonction `mysqli_stmt_result_metadata` retourne NULL .

8.81.103 `mysqli_store_result()` : `mysqli->store_result()`

Style procédural

`mysqli_result` **`mysqli_store_result`** (`mysqli link`)

Style orienté objet (méthode)

`mysqli` `mysqli_result` **`store_result`** ()

Transfère le jeu de résultats à partir de la dernière requête sur la connexion à la base de données spécifiée par le paramètre `link` pour une utilisation avec `mysqli_data_seek` .

Note

Il est toujours conseillé de libérer la mémoire allouée pour le résultat en utilisant la fonction `mysqli_free_result` , lors du transfert de grands résultats en utilisant la fonction `mysqli_store_result` cela devient particulièrement important.

Note

`mysqli_store_result` retourne FALSE dans le cas où la requête ne retourne pas de jeu de résultat (si la requête est de type INSERT par exemple). Cette fonction retournera toujours FALSE si le jeu de résultats ne peut être lu. Vous pouvez savoir s'il y a une erreur en utilisant la fonction `mysqli_error` et en regardant si elle renvoie une chaîne vide, ou si `mysqli_errno` retourne zéro, ou bien si `mysqli_field_count` retourne une valeur différente de zéro. Une autre raison pour que cette fonction retourne FALSE est que le jeu de résultats retourné après une requête réussie appelée par `mysqli_query` est trop long (la mémoire pour celui-ci ne peut être allouée). Si `mysqli_field_count` retourne une valeur différente de zéro, le traitement devrait produire un jeu de résultats non vide.

8.81.104 `mysqli_thread_id()` : `mysqli->thread_id()`

Style procédural

int **`mysqli_thread_id`** (`mysqli link`)

Style orienté objet (méthode)

`mysqli` int `thread_id`

La fonction `mysqli_thread_id` retourne l'identifiant du thread de la connexion courante qui peut être terminé par la suite en utilisant la fonction `mysqli_kill` .

Note

L'identifiant du thread est assigné sur une base de connexion par connexion. Ce qui fait que si la connexion est coupée, puis relancée, un nouvel identifiant de thread lui sera assigné.

Pour terminer une requête en cours d'exécution, vous pouvez utiliser la commande SQL KILL QUERY processid .

8.81.105 mysqli_thread_safe() : Indique si le thread safety est donné ou non

Style procédural

bool **mysqli_thread_safe** ()

mysqli_thread_safe indique si la bibliothèque cliente est compilée avec le thread safety.

8.81.106 mysqli_use_result() : mysqli->use_result()

Style procédural

mysqli_result **mysqli_use_result** (mysqli link)

Style orienté objet (méthode)

mysqli mysqli_result **use_result** ()

mysqli_use_result est utilisée pour initialiser la récupération d'un jeu de résultats à partir de la dernière requête exécutée en utilisant la fonction mysqli_real_query sur la connexion spécifiée par le paramètre link . Cette fonction ou la fonction mysqli_store_result doivent être appelées avant que le résultat d'une requête ne puisse être récupéré, et pour éviter l'echec de la prochaine requête sur la connexion à la base de données.

Note

La fonction mysqli_use_result ne transfère pas le jeu de résultats en entier à partir de la base de données et on ne peut donc pas utiliser des fonctions telle mysqli_data_seek pour se déplacer entre les enregistrements. Pour utiliser cette fonctionnalité, vous devez récupérer le jeu de résultats en utilisant mysqli_store_result .

8.81.107 mysqli_warning_count() : mysqli->warning_count()

Style procédural

int **mysqli_warning_count** (mysqli link)

Style orienté objet (méthode)

mysqli int warning_count

mysqli_warning_count retourne le nombre d'avertissements générés par la dernière requête de la connexion représentée par le paramètre link .

Note

Pour récupérer les messages d'avertissements, vous pouvez utiliser la commande SQL `SHOW WARNINGS [limit row_count]` .

8.82 Fonctions de contrôle d'écran de terminal

8.82.1 Introduction

ncurses (new curses, les nouveaux curseurs) sont une émulation libre des curseurs du System V Rel 4.0 (et plus récente). Ils utilisent le format terminfo, supportent les pavés numériques, les couleurs, les colorations multiples, les caractères de formulaire et les touches de fonctions. De par la nature interactive de cette bibliothèque, il est hors de question de l'utiliser pour écrire des applications pour le Web mais, par contre, elle peut être très intéressante pour écrire des scripts en ligne de commande .

Attention

Ce module est **EXPERIMENTAL** . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, **SANS PREAVIS!** Soyez-en conscient, et utilisez ce module à vos risques et périls.

Ncurses est disponible sur les plates-formes suivantes :

- AIX
- BeOS
- Cygwin
- Digital Unix (aka OSF1)
- FreeBSD
- GNU/Linux
- HPUX
- IRIX
- OS/2
- SCO OpenServer
- Solaris
- SunOS

8.82.2 Pré-requis

Vous devez disposer des bibliothèques ncurses et des fichiers d'en-têtes. Téléchargez la dernière version sur le site de <ftp://ftp.gnu.org/pub/gnu/ncurses/> ou sur un miroir GNU.

8.82.3 Installation

Pour installer cette extension, compilez PHP en mode CGI ou CLI, avec l'option `--with-ncurses[=DIR]` .

8.82.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable	Historique
<code>ncurses.value</code>	"42"	PHP_INI_ALL	
<code>ncurses.string</code>	"foobar"	PHP_INI_ALL	

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#) .

8.82.5 Types de ressources

Cette extension ne définit aucune ressource.

8.82.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

8.82.6.1 Codes d'erreurs

En cas d'erreur, les fonctions ncurses retournent `NCURSES_ERR`.

8.82.6.2 Colors

Constante	Signification
<code>NCURSES_COLOR_BLACK</code>	Pas de couleur
<code>NCURSES_COLOR_WHITE</code>	blanc
<code>NCURSES_COLOR_RED</code>	rouge - supporté lorsque le terminal est en mode couleurs
<code>NCURSES_COLOR_GREEN</code>	vert - supporté lorsque le terminal est en mode couleurs
<code>NCURSES_COLOR_YELLOW</code>	jaune - supporté lorsque le terminal est en mode couleurs
<code>NCURSES_COLOR_BLUE</code>	bleu - supporté lorsque le terminal est en mode couleurs
<code>NCURSES_COLOR_CYAN</code>	cyan - supporté lorsque le terminal est en mode couleurs
<code>NCURSES_COLOR_MAGENTA</code>	magenta - supporté lorsque le terminal est en mode couleurs

8.82.6.3 Touches

Constante	Signification
<code>NCURSES_KEY_F0 - NCURSES_KEY_F64</code>	Touches de fonctions F1 - F64
<code>NCURSES_KEY_DOWN</code>	flèche vers le bas
<code>NCURSES_KEY_UP</code>	flèche vers le haut
<code>NCURSES_KEY_LEFT</code>	flèche vers la gauche
<code>NCURSES_KEY_RIGHT</code>	flèche vers la droite
<code>NCURSES_KEY_HOME</code>	touche home (vers le haut + flèche vers la gauche)
<code>NCURSES_KEY_BACKSPACE</code>	retour en arrière
<code>NCURSES_KEY_DL</code>	efface une ligne
<code>NCURSES_KEY_IL</code>	insère une ligne
<code>NCURSES_KEY_DC</code>	efface un caractère
<code>NCURSES_KEY_IC</code>	insère un caractère, ou bien passe en mode insertion
<code>NCURSES_KEY_EIC</code>	termine le mode insertion
<code>NCURSES_KEY_CLEAR</code>	efface l'écran
<code>NCURSES_KEY_EOS</code>	efface l'écran jusqu'en bas
<code>NCURSES_KEY_EOL</code>	efface l'écran jusqu'à la fin de la ligne
<code>NCURSES_KEY_SF</code>	scrolle une ligne vers le bas

NCURSES_KEY_SR	scrolle une ligne vers le haut
NCURSES_KEY_NPAGE	page suivante
NCURSES_KEY_PPAGE	page précédente
NCURSES_KEY_STAB	place une tabulation
NCURSES_KEY_CTAB	supprime une tabulation
NCURSES_KEY_CATAB	supprime toutes les tabulations
NCURSES_KEY_SRESET	remise à zéro partielle
NCURSES_KEY_RESET	remise à zéro totale
NCURSES_KEY_PRINT	imprime
NCURSES_KEY_LL	inférieur gauche (sur le pavé numérique)
NCURSES_KEY_A1	supérieur gauche (sur le pavé numérique)
NCURSES_KEY_A3	supérieur droit (sur le pavé numérique)
NCURSES_KEY_B2	centre (sur le pavé numérique)
NCURSES_KEY_C1	inférieur gauche (sur le pavé numérique)
NCURSES_KEY_C3	inférieur droit (sur le pavé numérique)
NCURSES_KEY_BTAB	tabulation arrière
NCURSES_KEY_BEG	début
NCURSES_KEY_CANCEL	annule
NCURSES_KEY_CLOSE	ferme
NCURSES_KEY_COMMAND	commande
NCURSES_KEY_COPY	copie
NCURSES_KEY_CREATE	crée
NCURSES_KEY_END	fin
NCURSES_KEY_EXIT	quitte
NCURSES_KEY_FIND	trouve
NCURSES_KEY_HELP	aide
NCURSES_KEY_MARK	marque
NCURSES_KEY_MESSAGE	message
NCURSES_KEY_MOVE	déplace
NCURSES_KEY_NEXT	suivant
NCURSES_KEY_OPEN	ouvre
NCURSES_KEY_OPTIONS	options
NCURSES_KEY_PREVIOUS	précédent
NCURSES_KEY_REDO	refaire
NCURSES_KEY_REFERENCE	référence
NCURSES_KEY_REFRESH	rafraîchis
NCURSES_KEY_REPLACE	remplace
NCURSES_KEY_RESTART	redémarre
NCURSES_KEY_RESUME	recommence
NCURSES_KEY_SAVE	sauve
NCURSES_KEY_SBEG	shift et début
NCURSES_KEY_SCANCEL	shift et annule
NCURSES_KEY_SCOMMAND	shift et commande
NCURSES_KEY_SCOPY	shift et copie
NCURSES_KEY_SCREATE	shift et crée

NCURSES_KEY_SDC	shift et efface un caractère
NCURSES_KEY_SDL	shift et efface une ligne
NCURSES_KEY_SELECT	sélectionne
NCURSES_KEY_SEND	shift et fin
NCURSES_KEY_SEOL	shift et fin de ligne
NCURSES_KEY_SEXIT	shift et quitte
NCURSES_KEY_SFIND	shift et trouve
NCURSES_KEY_SHELP	shift et aide
NCURSES_KEY_SHOME	shift et home
NCURSES_KEY_SIC	shift et entrée
NCURSES_KEY_SLEFT	shift et flèche vers la gauche
NCURSES_KEY_SMESSAGE	shift et message
NCURSES_KEY_SMOVE	shift et déplace
NCURSES_KEY_SNEXT	shift et suivant
NCURSES_KEY_SOPTIONS	shift et options
NCURSES_KEY_SPREVIOUS	shift et précédent
NCURSES_KEY_SPRINT	shift et imprime
NCURSES_KEY_SREDO	shift et refait
NCURSES_KEY_SREPLACE	shift et remplace
NCURSES_KEY_SRIGHT	shift et flèche vers la droite
NCURSES_KEY_SRSUME	shift et recommence
NCURSES_KEY_SSAVE	shift et sauve
NCURSES_KEY_SSUSPEND	shift et suspend
NCURSES_KEY_UNDO	défait
NCURSES_KEY_MOUSE	un événement souris est survenu
NCURSES_KEY_MAX	valeur maximale de clé

Souris

Constante	Signification
NCURSES_BUTTON1_RELEASED - NCURSES_BUTTON4_RELEASED	bouton (1-4) relaché
NCURSES_BUTTON1_PRESSED - NCURSES_BUTTON4_PRESSED	bouton (1-4) pressé
NCURSES_BUTTON1_CLICKED - NCURSES_BUTTON4_CLICKED	bouton (1-4) cliqué
NCURSES_BUTTON1_DOUBLE_CLICKED - NCURSES_BUTTON4_DOUBLE_CLICKED	bouton (1-4) double cliqué
NCURSES_BUTTON1_TRIPLE_CLICKED - NCURSES_BUTTON4_TRIPLE_CLICKED	bouton (1-4) triple cliqué
NCURSES_BUTTON_CTRL	ctrl pressé durant le clic
NCURSES_BUTTON_SHIFT	shift pressé durant le clic
NCURSES_BUTTON_ALT	alt pressé durant le clic
NCURSES_ALL_MOUSE_EVENTS	indique tous les événements souris
NCURSES_REPORT_MOUSE_POSITION	indique la position de la souris

- [Introduction](#)

- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [ncurses_addch](#)
- [ncurses_addchnstr](#)
- [ncurses_addchstr](#)
- [ncurses_addnstr](#)
- [ncurses_addstr](#)
- [ncurses_assume_default_colors](#)
- [ncurses_attroff](#)
- [ncurses_atron](#)
- [ncurses_attrset](#)
- [ncurses_baudrate](#)
- [ncurses_beep](#)
- [ncurses_bkgd](#)
- [ncurses_bkgdset](#)
- [ncurses_border](#)
- [ncurses_bottom_panel](#)
- [ncurses_can_change_color](#)
- [ncurses_cbreak](#)
- [ncurses_clear](#)
- [ncurses_clrtoebot](#)
- [ncurses_clrtoeol](#)
- [ncurses_color_content](#)
- [ncurses_color_set](#)
- [ncurses_curs_set](#)
- [ncurses_def_prog_mode](#)
- [ncurses_def_shell_mode](#)
- [ncurses_define_key](#)
- [ncurses_del_panel](#)
- [ncurses_delay_output](#)
- [ncurses_delch](#)
- [ncurses_deleteln](#)
- [ncurses_delwin](#)
- [ncurses_doupdate](#)
- [ncurses_echo](#)
- [ncurses_echochar](#)
- [ncurses_end](#)
- [ncurses_erase](#)
- [ncurses_erasechar](#)
- [ncurses_filter](#)
- [ncurses_flash](#)
- [ncurses_flushinp](#)
- [ncurses_getch](#)
- [ncurses_getmaxyx](#)
- [ncurses_getmouse](#)
- [ncurses_getyx](#)
- [ncurses_halfdelay](#)
- [ncurses_has_colors](#)
- [ncurses_has_ic](#)
- [ncurses_has_il](#)
- [ncurses_has_key](#)

- [ncurses_hide_panel](#)
- [ncurses_hline](#)
- [ncurses_inch](#)
- [ncurses_init_color](#)
- [ncurses_init_pair](#)
- [ncurses_init](#)
- [ncurses_insch](#)
- [ncurses_insdelln](#)
- [ncurses_insertln](#)
- [ncurses_insstr](#)
- [ncurses_instr](#)
- [ncurses_isendwin](#)
- [ncurses_keyok](#)
- [ncurses_keypad](#)
- [ncurses_killchar](#)
- [ncurses_longname](#)
- [ncurses_meta](#)
- [ncurses_mouse_trafo](#)
- [ncurses_mouseinterval](#)
- [ncurses_mousemask](#)
- [ncurses_move_panel](#)
- [ncurses_move](#)
- [ncurses_mvaddch](#)
- [ncurses_mvaddchnstr](#)
- [ncurses_mvaddchstr](#)
- [ncurses_mvaddnstr](#)
- [ncurses_mvaddstr](#)
- [ncurses_mvcur](#)
- [ncurses_mvdelch](#)
- [ncurses_mvgetch](#)
- [ncurses_mvhline](#)
- [ncurses_mvinch](#)
- [ncurses_mvvline](#)
- [ncurses_mvwaddstr](#)
- [ncurses_napms](#)
- [ncurses_new_panel](#)
- [ncurses_newpad](#)
- [ncurses_newwin](#)
- [ncurses_nl](#)
- [ncurses_nocbreak](#)
- [ncurses_noecho](#)
- [ncurses_nonl](#)
- [ncurses_noqiflush](#)
- [ncurses_noraw](#)
- [ncurses_pair_content](#)
- [ncurses_panel_above](#)
- [ncurses_panel_below](#)
- [ncurses_panel_window](#)
- [ncurses_pnoutrefresh](#)
- [ncurses_prefresh](#)
- [ncurses_putp](#)
- [ncurses_qiflush](#)
- [ncurses_raw](#)
- [ncurses_refresh](#)

- [ncurses_replace_panel](#)
- [ncurses_reset_prog_mode](#)
- [ncurses_reset_shell_mode](#)
- [ncurses_resetty](#)
- [ncurses_savetty](#)
- [ncurses_scr_dump](#)
- [ncurses_scr_init](#)
- [ncurses_scr_restore](#)
- [ncurses_scr_set](#)
- [ncurses_scr1](#)
- [ncurses_show_panel](#)
- [ncurses_slk_attr](#)
- [ncurses_slk_attroff](#)
- [ncurses_slk_attron](#)
- [ncurses_slk_attrset](#)
- [ncurses_slk_clear](#)
- [ncurses_slk_color](#)
- [ncurses_slk_init](#)
- [ncurses_slk_noutrefresh](#)
- [ncurses_slk_refresh](#)
- [ncurses_slk_restore](#)
- [ncurses_slk_set](#)
- [ncurses_slk_touch](#)
- [ncurses_standend](#)
- [ncurses_standout](#)
- [ncurses_start_color](#)
- [ncurses_termattrs](#)
- [ncurses_termname](#)
- [ncurses_timeout](#)
- [ncurses_top_panel](#)
- [ncurses_typeahead](#)
- [ncurses_ungetch](#)
- [ncurses_ungetmouse](#)
- [ncurses_update_panels](#)
- [ncurses_use_default_colors](#)
- [ncurses_use_env](#)
- [ncurses_use_extended_names](#)
- [ncurses_vidattr](#)
- [ncurses_vline](#)
- [ncurses_waddch](#)
- [ncurses_waddstr](#)
- [ncurses_wattroff](#)
- [ncurses_wattron](#)
- [ncurses_wattrset](#)
- [ncurses_wborder](#)
- [ncurses_wclear](#)
- [ncurses_wcolor_set](#)
- [ncurses_werase](#)
- [ncurses_wgetch](#)
- [ncurses_whline](#)
- [ncurses_wmouse_trafo](#)
- [ncurses_wmove](#)
- [ncurses_wnoutrefresh](#)
- [ncurses_wrefresh](#)

- [ncurses_wstandend](#)
- [ncurses_wstandout](#)
- [ncurses_wvline](#)

8.82.8 `ncurses_addchnstr()` : Ajoute une chaîne de taille donnée à la position courante

int `ncurses_addchnstr` (string s , int n)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.9 `ncurses_addchstr()` : Ajoute une chaîne à la position courante

int `ncurses_addchstr` (string s)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.10 `ncurses_addnstr()` : Ajoute une chaîne à la position courante

int `ncurses_addnstr` (string s , int n)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.11 `ncurses_addstr()` : Affiche du texte à la position courante

int `ncurses_addstr` (string text)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.12 ncurses_assume_default_colors() : Définit la couleur 0

int **ncurses_assume_default_colors** (int fg , int bg)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.13 ncurses_attroff() : Désactive les attributs donnés

int **ncurses_attroff** (int attributes)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.14 ncurses_attron() : Active les attributs suivants

int **ncurses_attron** (int attributes)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.15 ncurses_attrset() : Modifie les attributs donnés

int **ncurses_attrset** (int attributes)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.16 ncurses_baudrate() : Retourne le baudrate du terminal

int `ncurses_baudrate` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.17 ncurses_beep() : Fait beeper le terminal

int `ncurses_beep` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_beep` envoie une alerte audible (bell) et si ce n'est pas possible, fait flasher l'écran. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `ncurses_flash` .

8.82.18 ncurses_bkgd() : Définit les propriétés de fond d'écran pour le terminal

int `ncurses_bkgd` (int attrchar)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.19 ncurses_bkgdset() : Définit le fond d'écran

void `ncurses_bkgdset` (int attrchar)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.20 `ncurses_border()` : Dessine un bord autour de l'écran avec les caractères donnés

int `ncurses_border` (int left , int right , int top , int bottom , int tl_corner , int tr_corner , int bl_corner , int br_corner)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_border` dessine les lignes spécifiées et les coins autour de la fenêtre principale. Chaque paramètre attend 0 pour dessiner la ligne et 1 pour ne pas le faire. Les coins sont en haut à gauche, en haut à droite, en bas à gauche et en bas à droite.

Utilisez `ncurses_wborder` pour les bordures autour de la fenêtre.

Voir aussi `ncurses_wborder` .

8.82.21 `ncurses_bottom_panel()` : Place un conteneur visible en bas de la pile

int `ncurses_bottom_panel` (resource panel)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.22 `ncurses_can_change_color()` : Vérifie si le terminal peut changer de couleurs

bool `ncurses_can_change_color` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_can_change_color` retourne TRUE ou FALSE , suivant que le terminal supporte les couleurs ou pas, et si le programmeur peut changer ces couleurs ou pas.

8.82.23 `ncurses_cbreak()` : Change la bufferisation d'entrée

bool `ncurses_cbreak` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_cbreak` désactive la bufferisation de ligne, et le traitement des caractères (les interruptions et le flux de caractères de contrôle sont intouchés), rendant les caractères tapés par l'utilisateur directement accessibles au programme.

`ncurses_cbreak` retourne TRUE en cas de succès, et `NCURSES_ERR` en cas d'erreur.

Voir aussi `ncurses_nocbreak` .

8.82.24 `ncurses_clear()` : Efface l'écran

bool `ncurses_clear` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_clear` efface totalement l'écran sans configurer les blancs. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note : `ncurses_clear` efface totalement l'écran sans configurer les blancs, qui représentent la couleur de fond d'écran. Pour effacer le fond d'écran avec les blancs, utilisez plutôt `ncurses_erase` .

Voir aussi `ncurses_erase` .

8.82.25 `ncurses_clrtoebot()` : Efface l'écran depuis la position courante jusqu'au bas de l'écran

bool `ncurses_clrtoebot` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_clrtoebot` efface toutes les lignes de l'écran depuis le curseur jusqu'à la fin de l'écran. Les blancs qui sont créés avec la fonction `ncurses_clrtoebot` ont le rendu courant du fond d'écran. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `ncurses_clear` et `ncurses_clrtoeol`

8.82.26 `ncurses_clrtoeol()` : Efface l'écran depuis la position courante jusqu'à la fin de la ligne

bool `ncurses_clrtoeol` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_clrtoeol` efface la ligne courante depuis la position courante du curseur jusqu'à la fin. Les blancs qui sont créés avec la fonction `ncurses_clrtoeol` ont le rendu courant du fond d'écran. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `ncurses_clear` et `ncurses_clrtoebot`

8.82.27 `ncurses_color_content()` : Lit la valeur RGB d'une couleur

int `ncurses_color_content` (int color , int r , int g , int b)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.28 `ncurses_color_set()` : Modifie la couleur de fond et de devant

int `ncurses_color_set` (int pair)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.29 `ncurses_curs_set()` : Modifie l'état du curseur

int `ncurses_curs_set` (int visibility)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.30 `ncurses_def_prog_mode()` : Sauve le mode du terminal

bool `ncurses_def_prog_mode` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_def_prog_mode` sauve le mode du terminal pour réutilisation (dans les curses) avec la fonction `ncurses_reset_prog_mode` . `ncurses_def_prog_mode` retourne FALSE en cas de succès, et TRUE sinon.

Voir aussi `ncurses_reset_prog_mode` .

8.82.31 `ncurses_def_shell_mode()` : Sauve le mode de terminal (shell)

bool `ncurses_def_shell_mode` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_def_shell_mode` sauve le mode de terminal courant pour qu'il soit réutilisé par le shell (et non pas par curses) avec la fonction `ncurses_reset_shell_mode` . `ncurses_def_shell_mode` retourne FALSE en cas de succès, et TRUE sinon.

Voir aussi `ncurses_reset_shell_mode` .

8.82.32 ncurses_define_key() : Défini un code de clé (keycode)

int `ncurses_define_key` (string definition , int keycode)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.33 ncurses_del_panel() : Retire un conteneur de la pile et l'efface (mais pas la fenêtre associée)

bool `ncurses_del_panel` (resource panel)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.34 ncurses_delay_output() : Retarde l'affichage sur les terminaux utilisant des caractères de remplissage

int `ncurses_delay_output` (int milliseconds)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.35 ncurses_delch() : Efface le caractère courant et décale le reste de la ligne vers la gauche

bool `ncurses_delch` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_delch` efface la caractère sous le curseur. Tous les caractères sont alors décalés vers la gauche sur la même ligne, et le dernier caractère est remplacé par un vide. La position du curseur ne change pas. `ncurses_delch` retourne FALSE en cas de succès, et TRUE sinon.

Voir aussi `ncurses_deleteln` .

8.82.36 `ncurses_deleteln()` : Efface la ligne courante, et monte l'écran d'une ligne

bool `ncurses_deleteln` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_deleteln` efface la ligne courante, qui est placée sous le curseur. Toutes les lignes sous-jacentes sont alors remontées d'un cran. La ligne du bas est effacée. La position du curseur ne change pas. `ncurses_deleteln` retourne FALSE en cas de succès, et sinon TRUE .

Voir aussi `ncurses_delch` .

8.82.37 `ncurses_delwin()` : Efface une fenêtre ncurses

bool `ncurses_delwin` (resource window)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.38 `ncurses_doupdate()` : Affiche tout ce qui est prêt, et rafraîchit l'écran

bool `ncurses_doupdate` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_doupdate` compare l'écran mémoire avec l'écran physique, et met à jour l'écran physique. Cette méthode est plus efficace que d'utiliser des rafraîchissements multiples. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.82.35 `ncurses_delch()` : Efface le caractère courant et décale le reste de la ligne vers la gauche

8.82.39 ncurses_echo() : Active l'écho d'entrée clavier

bool `ncurses_echo` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_echo` active le mode écho. Tous les caractères saisis par l'utilisateur sont affichés à l'écran par `ncurses_getch` . `ncurses_echo` retourne FALSE en cas de succès, et TRUE si une erreur survient.

Pour désactiver le mode écho, utilisez `ncurses_noecho` .

8.82.40 ncurses_echochar() : Affiche un caractère et rafraîchit l'écran

int `ncurses_echochar` (int character)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.41 ncurses_end() : Cesse l'utilisation de ncurses et efface l'écran

int `ncurses_end` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.42 ncurses_erase() : Efface l'écran du terminal

bool `ncurses_erase` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_erase` remplit le terminal avec des blancs. Les blancs créés ont la couleur du fond d'écran, configurée avec la fonction `ncurses_bkgd` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [ncurses_bkgd](#) et [ncurses_clear](#)

8.82.43 `ncurses_erasechar()` : Lit le caractère qui se fait effacer

string `ncurses_erasechar` (void)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_erasechar` retourne le caractère qui se fait effacer.

Voir aussi [ncurses_killchar](#) .

8.82.44 `ncurses_filter()` : Définit LINES pour `iniscr()` et `newterm()` à 1

void `ncurses_filter` (void)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.45 `ncurses_flash()` : Fait flasher le terminal (visual bell)

bool `ncurses_flash` (void)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_flash` fait flasher le terminal et si ce n'est pas possible, envoie une alerte sonore (bell).
`ncurses_flash` retourne FALSE en cas de succès, et TRUE sinon.

Voir aussi [ncurses_beep](#) .

8.82.46 `ncurses_flushinp()` : Vide le buffer d'entrée du clavier

bool `ncurses_flushinp` (void)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_flushinp` vide le buffer d'entrée du clavier que votre programme n'a pas encore lu. `ncurses_flushinp` retourne FALSE en cas de succès, et TRUE sinon.

8.82.47 `ncurses_getch()` : Lit un caractère sur le clavier

int `ncurses_getch` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.48 `ncurses_getmaxyx()` : Retourne la taille d'une fenêtre

void `ncurses_getmaxyx` (resource window , int y , int x)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

`ncurses_getmaxyx` place la taille verticale et la taille horizontale de la fenêtre window dans respectivement les variables y et x . Les variables doivent être passées par référence, donc, elles seront modifiées lorsque l'utilisateur change la taille du terminal.

8.82.49 `ncurses_getmouse()` : Lit les événements souris

bool `ncurses_getmouse` (array mevent)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_getmouse` lit les événements souris placés dans la queue. `ncurses_getmouse` retournera FALSE si un événement souris est visible dans la fenêtre, et sinon, elle retournera TRUE . Les options d'événements seront placées dans le paramètre mevent , qui doit être un tableau, passé par référence. En cas de succès, un tableau associatif avec les clés suivantes sera fourni :

- "id" : Identifiant permettant de distinguer plusieurs périphériques
- "x" : Position à l'écran, en abscisse relative, et comptée en caractères
- "y" : Position à l'écran, en ordonnée relative, et comptée en caractères
- "z" : Actuellement non supporté
- "mmask" : Action de souris

Exemple avec `ncurses_getmouse`

```
<?php
switch (ncurses_getch()){
    case NCURSES_KEY_MOUSE:
        if (!ncurses_getmouse(&$mevent)){
            if ($mevent["mmask"] & NCURSES_MOUSE_BUTTON1_PRESSED){
                $mouse_x = $mevent["x"]; // Sauve la position de la souris
            }
        }
    }
}
```

```

        $mouse_y = $mevent["y"];
    }
}
break;

default:
    /* .... */
}
?>

```

Voir aussi [ncurses_ungetmouse](#) .

8.82.50 `ncurses_getyx()` : Retourne la position courante du curseur pour une fenêtre

void `ncurses_getyx` (resource window , int y , int x)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.51 `ncurses_halfdelay()` : Met le terminal en mode semi-retardé

int `ncurses_halfdelay` (int tenth)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.52 `ncurses_has_colors()` : Vérifie que le terminal supporte les couleurs

bool `ncurses_has_colors` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_has_colors` retourne TRUE ou FALSE suivant que le terminal supporte les couleurs ou pas.

Voir aussi [ncurses_can_change_color](#) .

8.82.53 `ncurses_has_ic()` : Vérifie les capacités d'insertion et d'effacement

bool `ncurses_has_ic` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_has_ic` vérifie les capacités d'insertion et d'effacement du terminal. `ncurses_has_ic` retourne TRUE lorsque le terminal a ces capacités et sinon FALSE .

Voir aussi `ncurses_has_il` .

8.82.54 `ncurses_has_il()` : Vérifie les capacités d'insertion et d'effacement

bool `ncurses_has_il` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_has_il` vérifie les capacités d'insertion et d'effacement du terminal. `ncurses_has_il` retourne TRUE si le terminal a ces capacités, et sinon FALSE

Voir aussi `ncurses_has_ic` .

8.82.55 `ncurses_has_key()` : Vérifie la présence des touches de fonctions sur le clavier

int `ncurses_has_key` (int keycode)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.56 `ncurses_hide_panel()` : Retire un conteneur de la pile, pour le rendre invisible

int `ncurses_hide_panel` (resource panel)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.57 `ncurses_hline()` : Dessine une ligne horizontale à la position courante, en utilisant un caractère et une taille maximale

int `ncurses_hline` (int charattr , int n)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.58 ncurses_inch() : Lit le caractère et ses attributs à la position courante

string **ncurses_inch** (void)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
<u>ncurses_inch</u> retourne le caractère de la position courante.

8.82.59 ncurses_init_color() : Configure une nouvelle valeur RGB pour une couleur

int **ncurses_init_color** (int color , int r , int g , int b)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.60 ncurses_init_pair() : Alloue une paire de couleur

int **ncurses_init_pair** (int pair , int fg , int bg)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.61 ncurses_init() : Initialise ncurses

void **ncurses_init** (void)

Attention

8.82.57 ncurses_hline() : Dessine une ligne horizontale à la position courante, en utilisant un caractère et un

<p>Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.</p>

ncurses_init initialise l'interface ncurses et doit être utilisé avant n'importe quelle autre fonction ncurses.

8.82.62 ncurses_insch() : Insère un caractère et décale le reste de la ligne vers la droite, y compris le caractère courant

int **ncurses_insch** (int character)

<p>Attention</p>

<p>Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.</p>

<p>Attention</p>

<p>Cette fonction n'est pas documentée et seule la liste des arguments est disponible.</p>
--

8.82.63 ncurses_insdelln() : Insère des lignes devant la ligne courante en scrollant vers le bas (des nombres négatifs donneront un scroll vers le haut)

int **ncurses_insdelln** (int count)

<p>Attention</p>

<p>Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.</p>

<p>Attention</p>

<p>Cette fonction n'est pas documentée et seule la liste des arguments est disponible.</p>
--

8.82.64 ncurses_insertln() : Insère une ligne et décale le reste de l'écran vers le bas

bool **ncurses_insertln** (void)

<p>Attention</p>

<p>Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.</p>

ncurses_insertln insère une nouvelle ligne au-dessus de la ligne courante. La ligne du bas de l'écran sera perdue.

8.82.65 `ncurses_insstr()` : Insère une chaîne à la position courante, et décale le reste de la chaîne à droite

int `ncurses_insstr` (string text)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.66 `ncurses_instr()` : Lit une chaîne depuis le terminal

int `ncurses_instr` (string buffer)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_instr` retourne le nombre de caractères lus depuis la position courante jusqu'à la fin de la ligne. `buffer` contient les caractères lus. Les attributs sont supprimés de ces caractères.

8.82.67 `ncurses_isendwin()` : Ncurses est en mode endwin, des affichages normaux sont possibles

bool `ncurses_isendwin` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_isendwin` retourne TRUE , si `ncurses_endwin` a été appelé sans aucun appel subséquent à `ncurses_wrefresh` , et sinon FALSE .

Voir aussi `ncurses_endwin` et `ncurses_wrefresh` .

8.82.68 `ncurses_keyok()` : Active ou désactive un code de clé (keycode)

int `ncurses_keyok` (int keycode , bool enable)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.69 `ncurses_keypad()` : Active ou désactive le keypad

int `ncurses_keypad` (resource window , bool bf)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.70 `ncurses_killchar()` : Retourne la ligne du caractère actuellement supprimé

string `ncurses_killchar` (void)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

`ncurses_killchar` retourne la ligne du caractère actuellement supprimé.

Voir aussi `ncurses_erasechar` .

8.82.71 `ncurses_longname()` : Retourne la description du terminal

string `ncurses_longname` (void)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

`ncurses_longname` retourne une description complète du terminal. La description est tronquée à 128 caractères. En cas d'erreur, `ncurses_longname` retourne NULL .

Voir aussi `ncurses_termname` .

8.82.72 `ncurses_meta()` : Active/désactive les informations de méta clé 8-bits

int `ncurses_meta` (resource window , bool 8bit)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.73 `ncurses_mouse_trafo()` : Transforme les coordonnées

bool `ncurses_mouse_trafo` (int y , int x , bool toscreen)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.74 `ncurses_mouseinterval()` : Configure les délais entre les clics de souris

int `ncurses_mouseinterval` (int milliseconds)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.75 `ncurses_mousemask()` : Configure les événements de souris à surveiller

int `ncurses_mousemask` (int newmask , int oldmask)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

`ncurses_mousemask` configure les événements de souris à surveiller. Par défaut, aucun événement souris n'est rapporté. La fonction `ncurses_mousemask` retourne un masque pour indiquer quels paramètres de `newmask` peut être rapporté. En cas d'échec complet, elle retourne 0. Dans le paramètre `oldmask` , qui est passé par référence `ncurses_mousemask` retourne la valeur du masque précédent. Les événements souris sont représentés par `NCURSES_KEY_MOUSE` dans la fonction `ncurses_wgetch` . Pour lire les données d'événements dans la queue, utilisez la fonction `ncurses_getmouse` .

Comme effet secondaire, utiliser la valeur de 0 avec le paramètre `newmask` désactive la souris. Lui donner une valeur positive non nulle réactive la souris.

Les options de masque de souris sont les constantes suivantes :

- `NCURSES_BUTTON1_PRESSED`
- `NCURSES_BUTTON1_RELEASED`
- `NCURSES_BUTTON1_CLICKED`
- `NCURSES_BUTTON1_DOUBLE_CLICKED`
- `NCURSES_BUTTON1_TRIPLE_CLICKED`
- `NCURSES_BUTTON2_PRESSED`
- `NCURSES_BUTTON2_RELEASED`
- `NCURSES_BUTTON2_CLICKED`
- `NCURSES_BUTTON2_DOUBLE_CLICKED`
- `NCURSES_BUTTON2_TRIPLE_CLICKED`
- `NCURSES_BUTTON3_PRESSED`
- `NCURSES_BUTTON3_RELEASED`
- `NCURSES_BUTTON3_CLICKED`
- `NCURSES_BUTTON3_DOUBLE_CLICKED`
- `NCURSES_BUTTON3_TRIPLE_CLICKED`

- NCURSES_BUTTON4_PRESSED
- NCURSES_BUTTON4_RELEASED
- NCURSES_BUTTON4_CLICKED
- NCURSES_BUTTON4_DOUBLE_CLICKED
- NCURSES_BUTTON4_TRIPLE_CLICKED
- NCURSES_BUTTON_SHIFT>
- NCURSES_BUTTON_CTRL
- NCURSES_BUTTON_ALT
- NCURSES_ALL_MOUSE_EVENTS
- NCURSES_REPORT_MOUSE_POSITION

Exemple avec `ncurses_mousemask`

```
<?php
$newmask = NCURSES_BUTTON1_CLICKED + NCURSES_BUTTON1_RELEASED;
$mask = ncurses_mousemask($newmask, &$oldmask);
if ($mask & $newmask){
    echo "Toutes les options de souris spécifiées sont supportées\n";
}
?>
```

Voir aussi `ncurses_getmouse` , `ncurses_ungetmouse` et `ncurses_getch` .

8.82.76 `ncurses_move_panel()` : Déplace un conteneur pour que son coin supérieur gauche soit aux coordonnées [startx, starty]

int `ncurses_move_panel` (resource panel , int startx , int starty)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.77 `ncurses_move()` : Déplace la position d'affichage

int `ncurses_move` (int y , int x)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.78 `ncurses_mvaddch()` : Déplace la position courante et ajoute un caractère

int `ncurses_mvaddch` (int y , int x , int c)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.79 ncurses_mvaddchnstr() : Déplace la position et ajoute la chaîne attribuée avec la taille donnée

```
int ncurses_mvaddchnstr ( int y , int x , string s , int n )
```

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.80 ncurses_mvaddchstr() : Déplace la position et ajoute une chaîne attribuée

```
int ncurses_mvaddchstr ( int y , int x , string s )
```

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.81 ncurses_mvaddnstr() : Déplace la position et ajoute une chaîne de taille donnée

```
int ncurses_mvaddnstr ( int y , int x , string s , int n )
```

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.82 ncurses_mvaddstr() : Déplace la position et ajoute une chaîne

```
int ncurses_mvaddstr ( int y , int x , string s )
```

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS

PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.83 `ncurses_mvcur()` : Déplace immédiatement le curseur

int `ncurses_mvcur` (int `old_y` , int `old_x` , int `new_y` , int `new_x`)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.84 `ncurses_mvdelch()` : Déplace la position et efface le caractère, puis décale le reste de la ligne à gauche

int `ncurses_mvdelch` (int `y` , int `x`)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.85 `ncurses_mvgetch()` : Déplace la position et lit le caractère à la nouvelle position

int `ncurses_mvgetch` (int `y` , int `x`)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.86 `ncurses_mvhline()` : Choisi une nouvelle position et dessine une ligne horizontale avec le caractère donné, et de taille maximale donnée

int `ncurses_mvhline` (int `y` , int `x` , int `attrchar` , int `n`)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.87 ncurses_mvinch() : Déplace la position et lit le caractère à la nouvelle position

int ncurses_mvinch (int y , int x)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.88 ncurses_mvline() : Modifie la position et dessine une ligne verticale avec le caractère donné, de taille maximale donnée

int ncurses_mvline (int y , int x , int attrchar , int n)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.89 ncurses_mvwaddstr() : Ajoute une chaîne à une nouvelle position dans la fenêtre

int ncurses_mvwaddstr (resource window , int y , int x , string text)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.90 ncurses_napms() : Fait une pause

int ncurses_napms (int milliseconds)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.91 `ncurses_new_panel()` : Crée un nouveau conteneur et l'associe à une fenêtre

resource `ncurses_new_panel` (resource window)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.92 `ncurses_newpad()` : Crée un nouveau pad (fenêtre)

resource `ncurses_newpad` (int rows , int cols)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.93 `ncurses_newwin()` : Crée une nouvelle fenêtre

resource `ncurses_newwin` (int rows , int cols , int y , int x)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_newwin` crée une nouvelle fenêtre pour y dessiner des éléments. Cette fenêtre peut être positionnée en utilisant `x` , `y` , `rows` et `cols` . Lors de la création de la nouvelle fenêtre, pensez à utiliser `ncurses_getmaxyx` pour repérer les espaces disponibles, sachant qu'un terminal est individuel et peut varier. La valeur retournée est une ressource ID utilisée pour différencier les différentes fenêtres.

8.82.94 `ncurses_nl()` : Convertit les nouvelles lignes et retours chariot en sauts de ligne

bool `ncurses_nl` (void)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.95 ncurses_nocbreak() : Passe le terminal en mode cookedbool `ncurses_nocbreak` (void)**Attention**

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_nocbreak` retourne le terminal en mode normal (cooked). Initialement, le terminal peut ou peut ne pas être en mode cbreak, comme mode hérité. Par conséquent, un program doit appeler `ncurses_cbreak` et `ncurses_nocbreak` explicitement. `ncurses_nocbreak` retourne FALSE en cas de succès, et TRUE sinon.

Voir aussi `ncurses_cbreak` .

8.82.96 ncurses_noecho() : Désactive l'écho clavierbool `ncurses_noecho` (void)**Attention**

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_noecho` annule l'effet d'écho des caractères tapés à l'écran. `ncurses_noecho` retourne FALSE en cas de succès, et TRUE sinon.

Voir aussi `ncurses_echo` et `ncurses_getch`

8.82.97 ncurses_nonl() : Ne convertit pas les nouvelles lignes et retours chariot en sauts de lignebool `ncurses_nonl` (void)**Attention**

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.98 ncurses_noqiflush() : Ne vide pas les buffers sur réception des caractères de signauxvoid `ncurses_noqiflush` (void)**Attention**

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS

PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.99 `ncurses_noraw()` : Déconfigure le mode brut du terminal

bool `ncurses_noraw` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_noraw` déconfigure le mode brut du terminal. Le mode brut est similaire au mode `cbreak`, car les caractères tapés sont immédiatement passés au programme. La différence est qu'en mode brut, les caractères d'interruption, de fin de programme, de suspension et de suivi du flux de caractères sont passés sans être interprétés. `ncurses_noraw` retourne FALSE en cas de succès, et TRUE sinon.

Voir aussi `ncurses_raw` , `ncurses_cbreak` et `ncurses_nocbreak`

8.82.100 `ncurses_pair_content()` : Retourne la valeur RGB d'une couleur

int `ncurses_pair_content` (int pair , int f , int b)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.101 `ncurses_panel_above()` : Retourne le conteneur au-dessus du conteneur

resource `ncurses_panel_above` (resource panel)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Si le conteneur est null, retourne le dernier conteneur de la pile.

8.82.102 `ncurses_panel_below()` : Retourne le conteneur sous le conteneur

resource `ncurses_panel_below` (resource panel)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Si le conteneur est null, le premier conteneur de la pile est retourné.

8.82.103 ncurses_panel_window() : Retourne la fenêtre associée avec un conteneur

resource ncurses_panel_window (resource panel)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.104 ncurses_pnoutrefresh() : Copie une région depuis un pad dans un écran virtuel

int ncurses_pnoutrefresh (resource pad , int pminrow , int pmincol , int sminrow , int smincol , int smaxrow , int smaxcol)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.105 ncurses_prefresh() : Copie une région depuis un pad dans l'écran virtuel

int ncurses_prefresh (resource pad , int pminrow , int pmincol , int sminrow , int smincol , int smaxrow , int smaxcol)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.106 ncurses_putp() : Applique l'information d'espacement à la chaîne et l'affiche

int ncurses_putp (string text)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.107 ncurses_qiflush() : Vide les buffers après détection d'un caractère de signal

void ncurses_qiflush (void)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.108 ncurses_raw() : Passe le terminal en mode brut (raw)

bool **ncurses_raw** (void)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

ncurses_raw place le terminal en mode brut. Le mode brut est similaire au mode cbreak, car les caractères tapés sont immédiatement passés au programme. La différence est qu'en mode brut, les caractères d'interruption, de fin de programme, de suspension et de suivi du flux de caractères sont passés sans être interprétés. ncurses_raw retourne FALSE en cas de succès, et TRUE sinon.

Voir aussi ncurses_noraw , ncurses_cbreak et ncurses_nocbreak

8.82.109 ncurses_refresh() : Rafraîchit l'écran

int **ncurses_refresh** (int ch)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.110 ncurses_replace_panel() : Remplace la fenêtre associée à un conteneur

int **ncurses_replace_panel** (resource panel , resource window)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.111 ncurses_reset_prog_mode() : Restaure le mode prog sauvé par def_prog_mode

int **ncurses_reset_prog_mode** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.112 `ncurses_reset_shell_mode()` : Restaure le mode shell, mode sauvé par `def_shell_mode`

int `ncurses_reset_shell_mode` (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.113 `ncurses_resetty()` : Rétablit le terminal sauvé

bool `ncurses_resetty` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_resetty` rétablit le terminal sauvé par la fonction `ncurses_savetty` . `ncurses_resetty` retourne toujours FALSE .

Voir aussi `ncurses_savetty` .

8.82.114 `ncurses_savetty()` : Sauve l'état du terminal

bool `ncurses_savetty` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_savetty` sauve l'état courant du terminal. L'état courant du terminal peut être rétabli avec la fonction `ncurses_resetty` . `ncurses_savetty` retourne toujours FALSE .

Voir aussi `ncurses_resetty` .

8.82.115 `ncurses_scr_dump()` : Sauve un écran dans un fichier

int `ncurses_scr_dump` (string filename)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.116 ncurses_scr_init() : Initialise un écran depuis un fichier de sauvegarde

int ncurses_scr_init (string filename)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.117 ncurses_scr_restore() : Rétablit un écran depuis un fichier de sauvegarde

int ncurses_scr_restore (string filename)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.118 ncurses_scr_set() : Configure un écran depuis un fichier de sauvegarde

int ncurses_scr_set (string filename)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.119 ncurses_sscr() : Scrolle le contenu de la fenêtre vers le haut ou le bas, sans changer la position courante

int ncurses_sscr (int count)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.120 `ncurses_show_panel()` : Place un conteneur invisible en haut de la pile, pour le rendre visible

```
int ncurses_show_panel ( resource panel )
```

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.121 `ncurses_slk_attr()` : Retourne l'attribut de la clé soft label courante

```
bool ncurses_slk_attr ( void )
```

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

`ncurses_slk_attr` retourne l'attribut de la clé soft label. `ncurses_slk_attr` retourne FALSE en cas de succès, et TRUE sinon.

8.82.122 `ncurses_slk_attroff()` : Désactive l'attribut donné pour les étiquettes des function-key (fonctions clés)

```
int ncurses_slk_attroff ( int intarg )
```

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.123 `ncurses_slk_atron()` : Active l'attribut donné pour les étiquettes des function-key (fonctions clés)

```
int ncurses_slk_atron ( int intarg )
```

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.124 `ncurses_slk_attrset()` : Définit les attributs donnés à une étiquette function-key

int `ncurses_slk_attrset` (int intarg)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.125 `ncurses_slk_clear()` : Efface les soft labels de l'écran

bool `ncurses_slk_clear` (void)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_slk_clear` efface les clés label keys de l'écran. `ncurses_slk_clear` retourne FALSE en cas de succès, et TRUE sinon.

8.82.126 `ncurses_slk_color()` : Configure la couleur des clés soft label

int `ncurses_slk_color` (int intarg)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.127 `ncurses_slk_init()` : Initialise les fonctions de clés soft label

bool `ncurses_slk_init` (int format)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_slk_init` doit être appelée avant que `ncurses_initscr` et `ncurses_newterm` soient utilisées. Si `ncurses_initscr` utilise une ligne de `stdscr` pour émuler les soft labels, alors le paramètre `format` détermine comment les labels seront arrangés sur l'écran. En donnant au paramètre `format` la valeur 0, vous utiliserez un arrangement de type 3-2-3; la valeur 1 indiquera un arrangement 4-4; la valeur 2 indiquera un mode 4-4-4, mais une ligne d'index sera créée en plus.

8.82.128 `ncurses_slk_noutrefresh()` : Copie toutes les clés soft label à l'écran virtuel

`bool ncurses_slk_noutrefresh (void)`

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.129 `ncurses_slk_refresh()` : Copie les clés soft label à l'écran

`bool ncurses_slk_refresh (void)`

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_slk_refresh` copie les clés soft label de l'écran virtuel vers l'écran physique.

`ncurses_slk_refresh` retourne FALSE en cas de succès, et TRUE sinon.

8.82.130 `ncurses_slk_restore()` : Rétablit les clés soft label

`bool ncurses_slk_restore (void)`

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_slk_restore` rétablit les clés soft label après l'utilisation de `ncurses_slk_clear` .

8.82.131 `ncurses_slk_set()` : Modifie les étiquettes de clé de fonction (function key labels)

`bool ncurses_slk_set (int labelnr , string label , int format)`

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.132 `ncurses_slk_touch()` : Force l'affichage lorsque `ncurses_slk_noutrefresh` est utilisé

`bool ncurses_slk_touch (void)`

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_slk_touch` force l'affichage de tous les soft labels dès la prochaine utilisation de `ncurses_slk_noutrefresh` .

8.82.133 `ncurses_standend()` : Cesse l'utilisation de l'attribut 'standout'

int `ncurses_standend` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.134 `ncurses_standout()` : Commence l'utilisation de l'attribut 'standout'

int `ncurses_standout` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.135 `ncurses_start_color()` : Active les couleurs

int `ncurses_start_color` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.136 `ncurses_termattrs()` : Retourne toutes les options supportées par le terminal, combinées ensemble par l'opérateur OR

bool `ncurses_termattrs` (void)

Attention

<p>Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.</p>

<p>Attention</p>

<p>Cette fonction n'est pas documentée et seule la liste des arguments est disponible.</p>
--

8.82.137 ncurses_termname() : Retourne le nom court du terminal

string **ncurses_termname** (void)

<p>Attention</p>

<p>Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.</p>

ncurses_termname retourne le nom court du terminal. Le nom court est tronqué à 14 caractères. En cas d'erreur, ncurses_termname retourne NULL .

Voir aussi ncurses_longname .

8.82.138 ncurses_timeout() : Configure le temps d'expiration des séquences clavier spéciales

void **ncurses_timeout** (int millisec)

<p>Attention</p>

<p>Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.</p>

<p>Attention</p>

<p>Cette fonction n'est pas documentée et seule la liste des arguments est disponible.</p>
--

8.82.139 ncurses_top_panel() : Place un conteneur visible sur le haut de la pile

int **ncurses_top_panel** (resource panel)

<p>Attention</p>

<p>Cette fonction n'est pas documentée et seule la liste des arguments est disponible.</p>
--

8.82.140 ncurses_typeahead() : Spécifie un autre descripteur de fichier pour la vérification à la volée des données saisies

int **ncurses_typeahead** (int fd)

<p>Attention</p>

<p>Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.</p>

8.82.136 ncurses_termattrs() : Retourne toutes les options supportées par le terminal, combinées ensemble

PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.141 `ncurses_ungetch()` : Remplace un caractère dans la queue d'entrée

int `ncurses_ungetch` (int keycode)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.142 `ncurses_ungetmouse()` : Ajoute un événement souris dans la queue

bool `ncurses_ungetmouse` (array mevent)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

`ncurses_getmouse` ajoute un événement souris dans la queue d'entrée, et associe cet événement avec l'état du terminal et les coordonnées relatives (en caractères) de la souris, spécifiés dans `mevent` . Les options d'événements sont spécifiées dans le tableau associatif `mevent` :

- "id" : identifiant permettant de distinguer plusieurs périphériques
- "x" : position à l'écran, en abscisse relative, et comptée en caractères
- "y" : position à l'écran, en ordonnée relative, et comptée en caractères
- "z" : actuellement non supporté
- "mmask" : action de souris

`ncurses_ungetmouse` retourne FALSE en cas de succès, et TRUE sinon.

Voir aussi `ncurses_getmouse` .

8.82.143 `ncurses_update_panels()` : Rafraîchit l'écran virtuel pour prendre en compte les relations entre les conteneurs de la pile

void `ncurses_update_panels` (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.144 `ncurses_use_default_colors()` : Assigne la couleur de terminal pour l'index -1

`bool ncurses_use_default_colors (void)`

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.145 `ncurses_use_env()` : Contrôle l'utilisation des informations d'environnement concernant la taille du terminal

`void ncurses_use_env (bool flag)`

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.146 `ncurses_use_extended_names()` : Contrôle l'utilisation des noms étendus dans les descriptions terminfo

`int ncurses_use_extended_names (bool flag)`

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.147 `ncurses_vidattr()` : Affiche la chaîne sur un terminal dans le mode d'attribut vidéo

`int ncurses_vidattr (int intarg)`

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.148 `ncurses_vline()` : Dessine une ligne verticale à la position courante en utilisant un caractère donné, pour une taille donnée

int `ncurses_vline` (int charattr , int n)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.149 `ncurses_waddch()` : Ajoute un caractère à la position courante, dans une fenêtre, et avance le curseur

int `ncurses_waddch` (resource window , int ch)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.150 `ncurses_waddstr()` : Affiche le texte à la position courante dans la fenêtre

int `ncurses_waddstr` (resource window , string str , int n)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.151 `ncurses_wattroff()` : Désactive les attributs d'une fenêtre

int `ncurses_wattroff` (resource window , int attrs)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.152 `ncurses_wattron()` : Efface les attributs de la fenêtre

int `ncurses_wattron` (resource window , int attrs)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.153 ncurses_wattrset() : Modifie les attributs d'une fenêtre

int **ncurses_wattrset** (resource window , int attrs)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.154 ncurses_wborder() : Dessine le bord d'une fenêtre avec un caractère qualifié

int **ncurses_wborder** (resource window , int left , int right , int top , int bottom , int tl_corner , int tr_corner , int bl_corner , int br_corner)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

ncurses_wborder dessine les lignes spécifiées et les coins autour de la fenêtre passée par window . Chaque paramètre attend 0 pour dessiner et 1 pour ne pas le faire. Les coins sont en haut à gauche, en haut à droite, en bas à gauche et en bas à droite.

Utilisez ncurses_border pour les bordures de la fenêtre principale.

Voir aussi ncurses_border .

8.82.155 ncurses_wclear() : Efface la fenêtre

int **ncurses_wclear** (resource window)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.156 ncurses_wcolor_set() : Modifie les paires de couleurs de la fenêtre

int **ncurses_wcolor_set** (resource window , int color_pair)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.157 ncurses_werase() : Efface le contenu de la fenêtre

int **ncurses_werase** (resource window)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.158 ncurses_wgetch() : Lit un caractère depuis le clavier (fenêtre)

int **ncurses_wgetch** (resource window)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.159 ncurses_wline() : Dessine une ligne horizontale à partir de la position courante, avec un caractère qualifié, et d'une taille maximale

int ncurses_wline (resource window , int charattr , int n)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.160 ncurses_wmouse_trafo() : Transforme les coordonnées d'une fenêtre

bool ncurses_wmouse_trafo (resource window , int y , int x , bool toscreen)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.161 ncurses_wmove() : Déplace la position d'affichage de la fenêtre

int ncurses_wmove (resource window , int y , int x)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.162 ncurses_wnoutrefresh() : Copie la fenêtre dans l'écran virtuel

int ncurses_wnoutrefresh (resource window)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.163 ncurses_wrefresh() : Rafraîchit la fenêtre sur l'écran du terminal

int ncurses_wrefresh (resource window)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.164 ncurses_wstandend() : Termine le mode standout de la fenêtre

int ncurses_wstandend (resource window)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.165 ncurses_wstandout() : Passe une fenêtre en mode standout

int ncurses_wstandout (resource window)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.82.166 ncurses_wvline() : Dessine une ligne verticale dans une fenêtre, à la position courante, en utilisant un caractère qualifié, avec une taille maximale

int ncurses_wvline (resource window , int charattr , int n)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.83 Réseau

8.83.1 Introduction

8.83.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.83.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.83.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable	Historique
<code>define_syslog_variables</code>	"0"	PHP_INI_ALL	

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#).

Voici un éclaircissement sur l'utilisation des directives de configuration.

`define_syslog_variables` booléen

Définit s'il faut ou pas définir les différentes variables de syslog, (e.g. `$LOG_PID`, `$LOG_CRON`, etc.). Eteindre cette fonctionnalité est recommandé pour les performances. Durant l'exécution, vous pouvez définir ces variables en appelant `define_syslog_variables`.

8.83.5 Types de ressources

Cette extension ne définit aucune ressource.

8.83.6 Constantes pré-définies

Les constantes listées ici sont toujours disponibles dans PHP.

Constante	Description
<code>LOG_CONS</code>	S'il y a une erreur lors de l'envoi de données au système d'historique, l'erreur sera écrite directement sur la console système.
<code>LOG_NDELAY</code>	Ouvre la connexion à l'historique immédiatement
<code>LOG_ODELAY</code>	(par défaut) retarde l'ouverture de la connexion jusqu'à ce que le premier message soit archivé
<code>LOG_NOWAIT</code>	
<code>LOG_PERROR</code>	Affiche aussi le message sur la sortie standard

LOG_PID	Ajoute le PID dans chaque message	
Constante	Description	
LOG_AUTH	sécurité/autorisation message (utilisez LOG_AUTHPRIV à la place sur les systèmes où cette constante n'est pas définie)	
LOG_AUTHPRIV	sécurité/autorisation message (privé)	
LOG_CRON	démon de temps (cron et at)	
LOG_DAEMON	autres démons systèmes	
LOG_KERN	messages noyau	
LOG_LOCAL0 ... LOG_LOCAL7	réservées pour une utilisation locale, elles ne sont pas disponibles sous Windows	
LOG_LPR	imprimante système	
LOG_MAIL	système de mail	
LOG_NEWS	système de news USENET	
LOG_SYSLOG	messages générés en interne par le démon syslogd	
LOG_USER	messages génériques utilisateurs	
LOG_UUCP	système UUCP	
Constante	Description	
LOG_EMERG	Le système est inutilisable	
LOG_ALERT	Une mesure doit être prise immédiatement	
LOG_CRIT	Conditions critiques	
LOG_ERR	Erreurs	
LOG_WARNING	Alertes	
LOG_NOTICE	Notes (normales mais significatives)	
LOG_INFO	Message d'informations	
LOG_DEBUG	Message de débogage	
Constante	Description	
DNS_A	Adresse IPv4	
DNS_MX	Champs serveur Mail (MX)	
DNS_CNAME	Champs alias (Nom canonique) (A)	
DNS_NS	Champs de serveur autorité (NS)	
DNS_PTR	Champs de pointeur (PTR)	
DNS_HINFO	Champs d'informations d'hôte (HINFO) (voir la Operating System Names pour la signification de ces valeurs)	
DNS_SOA	Champs de délégation d'autorité (SOA)	
DNS_TXT	Texte	
DNS_ANY	Tous les champs. Sur la plupart des systèmes, cela retourne tous les champs, mais il ne faut pas s'y fier. Utilisez DNS_ALL .	
DNS_AAAA	Adresse IPv6	
DNS_ALL	Sollicite itérativement le serveur pour chaque type de champ.	

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [checkdnsrr](#)

- [closelog](#)
- [debugger_off](#)
- [debugger_on](#)
- [define_syslog_variables](#)
- [dns_check_record](#)
- [dns_get_mx](#)
- [dns_get_record](#)
- [fsockopen](#)
- [gethostbyaddr](#)
- [gethostbyname](#)
- [gethostbyname1](#)
- [getmxrr](#)
- [getprotobyname](#)
- [getprotobynumber](#)
- [getservbyname](#)
- [getservbyport](#)
- [inet_ntop](#)
- [inet_pton](#)
- [ip2long](#)
- [long2ip](#)
- [openlog](#)
- [pfsockopen](#)
- [socket_get_status](#)
- [socket_set_blocking](#)
- [socket_set_timeout](#)
- [syslog](#)

8.83.8 closelog() : Ferme la connexion à l'historique système

bool **closelog** (void)

[closelog](#) ferme le pointeur qui sert à écrire dans l'historique système. L'utilisation de [closelog](#) est optionnelle.

Voir aussi [define_syslog_variables](#) , [syslog](#) et [openlog](#) .

8.83.9 debugger_off() : Désactive le débogueur interne de PHP (PHP 3)

int **debugger_off** (void)

[debugger_off](#) désactive le débogueur interne de PHP. Cette fonction n'est valide qu'en PHP 3.

Pour plus d'informations, voir l'annexe sur le [débogueur PHP](#) .

8.83.10 debugger_on() : Active le débogueur interne de PHP (PHP 3)

int **debugger_on** (string address)

[debugger_on](#) active le débogueur interne de PHP, et le connecte à l'adresse `address` . Cette fonction n'est valide qu'en PHP 3.

Pour plus d'informations, voir l'annexe sur le [debugueur PHP](#) .

8.83.11 `define_syslog_variables()` : Initialise toutes les constantes liées au syslog

void `define_syslog_variables` (void)

`define_syslog_variables` initialise toutes les constantes utilisées par les fonctions de syslog.

Voir aussi [openlog](#) , [syslog](#) et [closelog](#) .

8.83.12 `dns_check_record()` : Synonyme de [checkdnsrr](#)

bool `dns_check_record` (string host , string type)

`dns_check_record` vérifie les enregistrements DNS correspondants à l'hôte internet host , donnés sous forme d'adresse IP ou de nom Internet.

8.83.13 `dns_get_mx()` : Synonyme de [getmxrr](#)

bool `dns_get_mx` (string hostname , array mxhosts , array weight)

`dns_get_mx` lit le champ MX de l'hôte internet hostname .

8.83.14 `dns_get_record()` : Lit les données DNS associées à un hôte

array `dns_get_record` (string hostname , int type , array authns , array addtl)

Note

`dns_get_record` n'est pas implémentée sur les plates-formes Windows ni sur les systèmes *BSD (actuellement). Utilisez la classe [PEAR Net_DNS](#) .

`dns_get_record` retourne un tableau associatif, contenant **au minimum** les index suivants :

Attribut	Signification
host	L'enregistrement de l'espace de nom DNS qui est décrit par les autres données.
class	<code>dns_get_record</code> ne retourne que la classe d'enregistrement Internet et, en tant que tel, cet index vaudra toujours IN .
type	Chaîne de caractère contenant le type d'enregistrement. Des attributs supplémentaires seront aussi disponibles dans le tableau suivant la valeur de ce type. Reportez-vous à la table ci-dessous.
ttl	Time To Live : Durée avant expiration de l'enregistrement. Cette valeur est différente de la durée avant expiration originale, mais plutôt cette valeur moins la durée depuis la dernière interrogation du serveur DNS responsable.

hostname doit être un nom d'hôte DNS valide, comme `www.example.com` . Des résolutions inversées peuvent être faites avec la notation `in-addr.arpa`, mais la fonction [gethostbyaddr](#) est plus efficace pour faire des résolutions inverses.

Par défaut, `dns_get_record` va rechercher toutes les ressources associées à `hostname` . Pour limiter la taille de la requête, le paramètre optionnel `type` peut prendre l'une des valeurs constantes suivantes : `DNS_A` , `DNS_CNAME` , `DNS_HINFO` , `DNS_MX` , `DNS_NS` , `DNS_PTR` , `DNS_SOA` , `DNS_TXT` , `DNS_AAAA` , `DNS_SRV` , `DNS_NAPTR` , `DNS_A6` , `DNS_ALL` ou `DNS_ANY` . Par défaut, il vaut **`DNS_ANY`** .

Note
A cause des performances excentriques de la bibliothèque <code>libresolv</code> , suivant les plates-formes, <code>DNS_ANY</code> ne retournera pas toujours tous les enregistrements, et l'option <code>DNS_ALL</code> , bien que plus lente, le fera de manière plus sûre.

Les paramètres optionnels `authns` et `adttl` sont passés par référence et, si fournis, ils recevront les enregistrements de ressources pour les **Authoritative Name Servers** , et tous les **Additional Records** respectivement. Voir l'exemple ci-dessous.

Type	Valeur supplémentaire
A	<code>ip</code> : une adresse IP v4, au format numérique.
MX	<code>pri</code> : priorité du serveur de mail. Les nombres faibles indiquent une priorité importante. <code>target</code> : FQDN du serveur de mail. Voir aussi <code>dns_get_mx</code> .
CNAME	<code>target</code> : FQDN du nom de l'espace DNS qui sert d'alias à cet enregistrement.
NS	<code>target</code> : FQDN du nom de serveur qui est responsable de ce nom de domaine.
PTR	<code>target</code> : nom de domaine vers lequel cet enregistrement pointe.
TXT	<code>txt</code> : chaîne de caractères arbitrairement associée à cet enregistrement.
HINFO	<code>cpu</code> : numéro IANA désignant le processeur de la machine référencée par cet enregistrement. <code>os</code> : numéro IANA désignant le système d'exploitation de la machine référencée par cet enregistrement. Voir Operating System Names pour connaître la signification de ces valeurs.
SOA	<code>mname</code> : FQDN de la source de cet enregistrement. <code>rname</code> : adresse email du contact administratif de ce domaine. <code>serial</code> : numéro de série du nom de domaine. <code>refresh</code> : intervalle de rafraîchissement (en secondes) que les serveurs de noms secondaires doivent utiliser pour mettre en cache ce nom de domaine. <code>retry</code> : durée (en secondes) d'attente après un rafraîchissement échoué, avant de faire une seconde tentative. <code>expire</code> : durée maximale (en secondes) de conservation d'une copie des données de zone sans pouvoir faire de rafraîchissement. <code>minimum-ttl</code> : durée minimale (en secondes) pendant laquelle un client conserve des données de zone avant qu'il ne soumette une nouvelle requête. Cette configuration peut être annulée par d'autres enregistrements.
AAAA	<code>ipv6</code> : adresse IPv6
A6 (PHP >= 5.1.0)	<code>masklen</code> : longueur (en octets) à hériter depuis la cible spécifiée par <code>chain</code> . <code>ipv6</code> : adresse pour que cet enregistrement spécifique fusionne avec <code>chain</code> . <code>chain</code> : l'enregistrement parent à fusionner avec les données <code>ipv6</code> .
SRV	<code>pri</code> : (priorité) les priorités les plus basses doivent être utilisées en premier. <code>weight</code> : classement pour choisir aléatoirement parmi les serveurs <code>targets</code> . <code>target</code> et <code>port</code> : nom d'hôte et port où le service est disponible. Pour plus d'informations, voir : RFC 2782
NAPTR	<code>order</code> et <code>pref</code> : équivalent à <code>pri</code> et <code>weight</code> ci-dessus. <code>flags</code> , <code>services</code> , <code>regex</code> , et <code>replacement</code> : paramètres tels que définis dans la RFC 2915 .

Note
En terme de standards DNS, les adresses email sont données au format <code>utilisateur.hote</code> (par exemple : <code>webmestre.example.com</code> au contraire du format <code>webmestre@example.com</code>). N'oubliez pas de vérifier cette adresse et de la modifier si nécessaire avant de la passer à la fonction <code>mail</code> .
Exemple avec <code>dns_get_record</code>

```
<?php
$result = dns_get_record("php.net");
print_r($result);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
  [0] => Array
  (
    [host] => php.net
    [type] => MX
    [pri] => 5
    [target] => pair2.php.net
    [class] => IN
    [ttl] => 6765
  )
  [1] => Array
  (
    [host] => php.net
    [type] => A
    [ip] => 64.246.30.37
    [class] => IN
    [ttl] => 8125
  )
)
*/
```

Comme il est très courant de rechercher l'IP d'un serveur, une fois que le champ MX a été résolu, `dns_get_record` retournera aussi un tableau dans le paramètre `addtl` qui contiendra les enregistrements associés. `authns` est aussi retourné en contenant une liste des serveurs autorité.

Exemple avec `dns_get_record` et `DNS_ANY`

```
<?php
/* Demande tous ("ANY") les enregistrements pour php.net,
   puis crée les tableaux $authns et $addtl
   contenant une liste des noms de serveurs, et tous
   les enregistrements qui vont avec
   */
$result = dns_get_record("php.net",DNS_ANY,$authns,$addtl);
echo 'Résultat = ';
print_r($result);
echo 'Autorité NS = ';
print_r($authns);
echo 'Additionnel = ';
print_r($addtl);
?>
```

L'exemple ci-dessus va afficher :

Résultat = Array

```
(
[0] => Array
(
[host] => php.net
[type] => MX
[pri] => 5
[target] => pair2.php.net
[class] => IN
[ttl] => 6765
)

[1] => Array
(
[host] => php.net
[type] => A
[ip] => 64.246.30.37
[class] => IN
[ttl] => 8125
)
)
Authortité NS = Array
(
[0] => Array
(
[host] => php.net
[type] => NS
[target] => remote1.easydns.com
[class] => IN
[ttl] => 10722
)

[1] => Array
(
[host] => php.net
[type] => NS
[target] => remote2.easydns.com
[class] => IN
[ttl] => 10722
)

[2] => Array
(
[host] => php.net
[type] => NS
[target] => ns1.easydns.com
[class] => IN
[ttl] => 10722
)

[3] => Array
(
[host] => php.net
[type] => NS
```

```
[target] => ns2.easydns.com
[class] => IN
[ttl] => 10722
)
)
Additionnel = Array
(
[0] => Array
(
[host] => pair2.php.net
[type] => A
[ip] => 216.92.131.5
[class] => IN
[ttl] => 6766
)

[1] => Array
(
[host] => remote1.easydns.com
[type] => A
[ip] => 64.39.29.212
[class] => IN
[ttl] => 100384
)

[2] => Array
(
[host] => remote2.easydns.com
[type] => A
[ip] => 212.100.224.80
[class] => IN
[ttl] => 81241
)

[3] => Array
(
[host] => ns1.easydns.com
[type] => A
[ip] => 216.220.40.243
[class] => IN
[ttl] => 81241
)

[4] => Array
(
[host] => ns2.easydns.com
[type] => A
[ip] => 216.220.40.244
[class] => IN
[ttl] => 81241
)
)
```

*/

Voir aussi [dns_get_mx](#) et [dns_check_record](#) .

8.83.15 fsockopen() : Ouvre une socket de connexion Internet ou Unix

resource **fsockopen** (string *target* , int *port* , int *errno* , string *errstr* , float *timeout*)

Initialise une connexion par socket à la ressource spécifiée par *target* . PHP supporte les cibles dans les domaines Internet et Unix comme décrit dans [Liste des modes de transports de sockets disponibles](#) . Une liste des types de transports peut également être trouvée en utilisant la fonction [stream_get_transports](#) .

Note

Si vous avez besoin de définir un délai limite pour lire/écrire des données à travers cette socket, utilisez la fonction [stream_set_timeout](#) , comme le paramètre *timeout* de la fonction [fsockopen](#) uniquement appliqué lors de la connexion de la socket.

Depuis PHP 4.3.0, si vous avez compilé le support OpenSSL, vous pouvez préfixer le paramètre *hostname* avec soit ' ssl://' soit ' tls://' pour utiliser une connexion SSL ou TLS avec le client à travers TCP/IP pour connecter l'hôte distant.

[fsockopen](#) retourne un pointeur de fichier qui peut être utilisé avec d'autres fonctions fichiers, telles [fgets](#) , [fgetss](#) , [fputs](#) , [fclose](#) et [feof](#) .

Si l'appel échoue, [fsockopen](#) retourne FALSE , et si les options *errno* et *errstr* ont été fournies, elles contiennent désormais les raisons de l'échec. Si l'erreur retournée est 0 et que la fonction retourne FALSE , c'est une indication d'erreur. C'est probablement dû à une erreur d'initialisation de la socket. Notez que *errno* et *errstr* sont toujours passées par référence.

Suivant les environnements, le type 'domaine Unix' ou l'option *timeout* ne sont pas toujours disponibles.

La socket sera ouverte par défaut en mode bloquant. Vous pouvez changer de mode en utilisant : [stream_set_blocking](#) .

Exemple avec [fsockopen](#)

```
<?php
$fp = fsockopen("www.example.com", 80, $errno, $errstr, 30);
if (!$fp) {
    echo "$errstr ($errno)<br />\n";
} else {
    $out = "GET / HTTP/1.1\r\n";
    $out .= "Host: www.example.com\r\n";
    $out .= "Connection: Close\r\n\r\n";

    fwrite($fp, $out);
    while (!feof($fp)) {
        echo fgets($fp, 128);
    }
    fclose($fp);
}
?>
```

L'exemple ci-dessous décrit comment lire la date et l'heure grâce à un service UDP "daytime" (port 13), sur votre propre machine.

Utilisation d'une connexion UDP

```
<?php
$fp = fsockopen("udp://127.0.0.1", 13, $errno, $errstr);
if (!$fp) {
    echo "ERREUR: $errno - $errstr<br />\n";
} else {
    fwrite($fp, "\n");
    echo fread($fp, 26);
    fclose($fp);
}
?>
```

Attention

Les sockets UDP semblent quelques fois avoir été ouvertes sans erreur, même si l'hôte distant n'est pas joignable. L'erreur apparaît alors uniquement lorsque vous tentez de lire/écrire sur la socket. La raison de cela est qu'UDP est un protocole "connectionless", ce qui signifie que le système ne tentera pas d'établir un lien pour la socket tant qu'il ne doit pas recevoir/envoyer de données.

Note

Lors de la spécification d'adresses IPv6 au format numérique (e.g. fe80::1) vous devez placer l'adresse IP entre crochets. Par exemple : tcp://[fe80::1]:80 .

Note

Le paramètre timeout a été introduit en PHP 3.0.9 et le support UDP en PHP 4.

Voir aussi [pfsockopen](#) , [socket_set_blocking](#) , [socket_set_timeout](#) , [fgets](#) , [fgetss](#) , [fwrite](#) , [fclose](#) , [feof](#) et l'extension [Curl](#) .

8.83.16 [gethostbyaddr\(\)](#) : Retourne le nom d'hôte correspondant à une IP

string [gethostbyaddr](#) (string ip_address)

[gethostbyaddr](#) retourne le nom d'hôte correspondant à l'IP ip_address . Si une erreur survient, retourne ip_address .

Exemple avec [gethostbyaddr](#)

```
<?php
$hostname = gethostbyaddr($_SERVER['REMOTE_ADDR']);

echo $hostname;
?>
```

Voir aussi [gethostbyname](#) et [gethostbyname1](#) .

8.83.17 [gethostbyname\(\)](#) : Retourne l'adresse IP correspondant à un hôte

string [gethostbyname](#) (string hostname)

[gethostbyname](#) retourne l'adresse IP correspondant à l'hôte hostname ou une chaîne correspondant à hostname en cas d'erreur.

Exemple avec [gethostbyaddr](#)

```
<?php
$ip = gethostbyname('www.example.com');
```

```
echo $ip;
?>
```

Voir aussi [gethostbyaddr](#) et [gethostbyname](#) .

8.83.18 [gethostbyname\(\)](#) : Retourne la liste d'IP correspondante à un hôte

array **gethostbyname** (string hostname)

[gethostbyname](#) retourne la liste d'IP correspondant à l'hôte hostname . Retourne FALSE si hostname ne peut être résolu.

Exemple avec [gethostbyname](#)

```
<?php
$notes = gethostbyname('www.yahoo.com');
print_r($notes);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[0] => 216.109.125.67
[1] => 64.58.76.228
[2] => 216.109.125.65
[3] => 216.109.125.70
[4] => 64.58.76.226
[5] => 64.58.76.230
[6] => 64.58.76.222
[7] => 216.109.125.66
)
```

Voir aussi [gethostbyname](#) , [gethostbyaddr](#) , [checkdnsrr](#) , [getmxrr](#) et la page 8 du manuel.

8.83.19 [getmxrr\(\)](#) : Retourne les enregistrements MX d'un hôte

bool **getmxrr** (string hostname , array mxhosts , array weight)

[getmxrr](#) effectue une recherche DNS pour obtenir les enregistrements MX de l'hôte hostname . Retourne TRUE si des enregistrements sont trouvés, et FALSE si une erreur est rencontrée, ou si la recherche échoue.

La liste des enregistrements MX est placée dans le tableau mxhosts . Si le tableau weight est fourni, il sera rempli par les informations de poids.

Note

Cette fonction ne doit pas être utilisée à des fins de vérification d'adresses. Seuls les serveurs de mails trouvés dans les enregistrements DNS sont retournés. Suivant la [RFC 2821](#) lorsqu'aucun serveur de mail n'est listé, hostname doit être utilisé comme serveur de mail, avec la priorité 0 .

Note

Cette fonction n'est pas disponible sur les machines Windows. Pour la remplacer, essayez la classe [PEAR](#) : [Net_DNS](#) .

Voir aussi [checkdnrr](#) , [dns_get_record](#) , [gethostbyname](#) , [gethostbyname1](#) , [gethostbyaddr](#) et la page 8 du manuel.

8.83.20 [getprotobyname\(\)](#) : Retourne le numéro de protocole associé à un nom de protocole

int [getprotobyname](#) (string name)

[getprotobyname](#) retourne le numéro de protocole associé avec le nom de protocole name , comme dans /etc/protocols .

Exemple avec [getprotobyname](#)

```
<?php
$protocol = 'tcp';
$get_prot = getprotobyname($protocol);
if ($get_prot == -1) {
    // if nothing found, returns -1
    echo 'Protocole invalide';
} else {
    echo 'Protocole #' . $get_prot;
}
?>
```

Voir aussi [getprotobynumber](#) .

8.83.21 [getprotobynumber\(\)](#) : Retourne le nom de protocole associé à un numéro de protocole

string [getprotobynumber](#) (int number)

[getprotobynumber](#) retourne le nom de protocole associé avec le numéro de protocole name , comme dans /etc/protocols .

Voir aussi [getprotobyname](#) .

8.83.22 [getservbyname\(\)](#) : Retourne le numéro de port associé à un service Internet et un protocole

int [getservbyname](#) (string service , string protocol)

[getservbyname](#) retourne le numéro de port associé au service service et au protocole protocol , comme dans /etc/services . protocol vaut soit tcp , soit udp (en minuscule). Retourne FALSE si service ou protocole n'est pas trouvé.

Exemple avec [getservbyname](#)

```
<?php
$services = array('http', 'ftp', 'ssh', 'telnet', 'imap',
'smtp', 'nickname', 'gopher', 'finger', 'pop3', 'www');
```

```

foreach ($services as $service) {
    $port = getservbyname($service, 'tcp');
    echo $service . ": " . $port . "<br />\n";
}
?>

```

Pour une liste complète des numéros de ports, voir :
<http://www.iana.org/assignments/port-numbers>.

Voir aussi [getservbyport](#) .

8.83.23 getservbyport() : Retourne le service Internet qui correspond au port et protocole

string **getservbyport** (int port , string protocol)

[getservbyport](#) le service internet associé au port port pour le protocole protocol comme dans /etc/services . protocol vaut soit tcp , soit udp (en minuscule).

Voir aussi [getservbyname](#) .

8.83.24 inet_ntop() : Convertit un paquet d'adresses internet en une représentation humainement lisible

string **inet_ntop** (string in_addr)

[inet_ntop](#) convertit une adresse 32bit IPv4 ou 128bit IPv6 (si PHP a été compilé avec le support IPv6) en une chaîne représentant une famille d'adresses. Retourne FALSE en cas d'échec.

Exemple avec [inet_ntop](#)

```

<?php
$packed = chr(127) . chr(0) . chr(0) . chr(1);
$expanded = inet_ntop($packed);

/* Affiche : 127.0.0.1 */
echo $expanded;

$packed = str_repeat(chr(0), 15) . chr(1);
$expanded = inet_ntop($packed);

/* Affiche : ::1 */
echo $expanded;
?>

```

Note

Cette fonction n'est pas implémentée sous Windows.

Voir aussi [long2ip](#) , [inet_pton](#) et [ip2long](#) .

8.83.25 inet_pton() : Convertit une adresse IP lisible en sa représentation in_addr

string **inet_pton** (string address)

8.83.22 [getservbyname\(\)](#) : Retourne le numéro de port associé à un service Internet et un protocole

`inet_pton` convertit une adresse IPv4 ou IPv6 (si PHP a été compilé avec le support IPv6) humainement lisible en une structure binaire appropriée de famille d'adresses 32bit ou 128bit.

Exemple avec `inet_pton`

```
<?php
$in_addr = inet_pton('127.0.0.1');

$in6_addr = inet_pton('::1');

?>
```

Note

Cette fonction n'est pas implémentée sous Windows.

Voir aussi `ip2long` , `inet_ntop` et `long2ip` .

8.83.26 `ip2long()` : Convertit une chaîne contenant une adresse (IPv4) IP numérique en adresse littérale

int `ip2long` (string ip_address)

`ip2long` génère une adresse IPv4 à partir de son équivalent numérique. Si `ip_address` est invalide, la fonction retournera -1 . Notez que -1 n'est pas évalué à FALSE en PHP.

Note

Depuis PHP 5.0.0, `ip2long` retourne FALSE si le paramètre `ip_address` n'est pas valide.

Exemple avec `ip2long`

```
<?php
$ip = gethostbyname('www.example.com');
$out = "Les URLs suivantes sont équivalentes :<br />\n";
$out .= 'http://www.example.com/, http://' . $ip . '/', et http://' . ip2long($ip) . "<br />\n";
echo $out;
?>
```

Note

Comme les entiers PHP sont signés et que beaucoup d'adresses IP peuvent être des entiers négatifs, vous devez utiliser le formateur "%u" de la fonction `sprintf` ou `printf` pour récupérer une représentation sous la forme d'une chaîne de caractères pour les adresses IP non signées.

Ce second exemple montre comment afficher une adresse convertie à l'aide de la fonction `printf` en PHP 4 et en PHP 5 :

Affichage d'adresse IP

```
<?php
$ip = gethostbyname('www.example.com');
$long = ip2long($ip);

if ($long == -1 || $long === FALSE) {
    echo 'IP invalide, merci d\'essayer encore';
} else {
    echo $ip . "\n";           // 192.0.34.166
    echo $long . "\n";        // -1073732954
    printf("%u\n", ip2long($ip)); // 3221234342
}
?>
```

`ip2long` ne devrait pas être utilisée comme seule méthode pour valider une adresse IP. Combinez-la avec `long2ip` :

Validation d'adresse IP

```
<?php
// s'assure que les IPs sont valides. Convertit également une IP incomplète
// en un format valide comme expliqué plus haut.
$ip = long2ip(ip2long("127.0.0.1")); // "127.0.0.1"
$ip = long2ip(ip2long("10.0.0")); // "10.0.0.0"
$ip = long2ip(ip2long("10.0.256")); // "10.0.1.0"
?>
```

`ip2long` fonctionne également avec des adresses IP incomplètes. Lisez http://publibn.boulder.ibm.com/doc_link/en_US/a_doc_lib/libs/commtrf2/inet_addr.htm pour plus d'informations.

Note

`ip2long` devrait retourner FALSE pour l'adresse IP 255.255.255.255 en PHP 5 <= 5.0.2. Ce comportement a été modifié en PHP 5.0.3 où il retournait -1 (comme en PHP 4).

Voir aussi `long2ip` et `sprintf` .

8.83.27 long2ip() : Convertit une adresse IP (IPv4) en adresse IP numérique

string **long2ip** (int proper_address)

`long2ip` génère une adresse IP (format aaa.bbb.ccc.ddd) à partir de sa représentation littérale.

Voir aussi `ip2long` .

8.83.28 openlog() : Ouvre la connexion à l'historique système

bool **openlog** (string ident , int option , int facility)

`openlog` ouvre la connexion à l'historique système. La chaîne `ident` sera ajoutée à chaque message. Les valeurs de `option` et `facility` sont données ci-dessous. `facility` sert à indiquer quel programme enregistre ce message. Cela vous permet de spécifier (sur la machine d'historique) comment traiter les messages venant de plusieurs serveurs. L'utilisation de `openlog` est optionnelle. Cette fonction sera automatiquement appelée par la fonction `syslog` si nécessaire, auquel cas `ident` vaudra par défaut FALSE .

Constante	Description
LOG_CONS	Si une erreur survient lors de l'envoi des données au gestionnaire d'historique, écrire directement l'erreur sur la console.
LOG_NDELAY	Ouvre immédiatement une connexion au gestionnaire d'historique
LOG_ODELAY	Retarde l'ouverture de la connexion jusqu'à ce que le premier message soit enregistré (par défaut)
LOG_PERROR	Envoie le message au gestionnaire standard
LOG_PID	Inclut le PID à chaque message

Vous pouvez utiliser une ou plusieurs de ces options. Pour les combiner, utiliser l'opérateur OR . Par exemple, pour ouvrir immédiatement la connexion, écrire sur la console et inclure le PID de

8.83.26 ip2long() : Convertit une chaîne contenant une adresse (IPv4) IPnumérique en adresse ~~872~~ littérale

chaque message, utilisez : LOG_CONS | LOG_NDELAY | LOG_PID .

Constante	Description
LOG_AUTH	sécurité/messages d'autorisation (utilisez LOG_AUTHPRIV, pour remplacer cette constante sur les systèmes où elle est définie).
LOG_AUTHPRIV	sécurité/messages d'autorisation (privé)
LOG_CRON	démon horloge (cron et at)
LOG_DAEMON	autres démons système
LOG_KERN	noyau (kernel)
LOG_LOCAL0 ... LOG_LOCAL7	réservé pour utilisation locale. Inutilisable sous Windows
LOG_LPR	imprimante (line printer subsystem)
LOG_MAIL	messagerie mail
LOG_NEWS	USENET : groupes de news (newsgroup)
LOG_SYSLOG	messages générés en interne par syslogd
LOG_USER	messages utilisateurs génériques
LOG_UUCP	UUCP subsystem
Note	
LOG_USER est le seul type valide sous les environnements Windows.	

Voir aussi [define_syslog_variables](#) , [syslog](#) et [closelog](#) .

8.83.29 pfsockopen() : Ouvre une socket de connexion Internet ou Unix persistante

resource **pfsockopen** (string hostname , int port , int errno , string errstr , float timeout)

[pfsockopen](#) se comporte exactement comme [fsockopen](#) mais la connexion ouverte le reste, même après la fin du script. C'est la version persistante de [fsockopen](#) .

8.83.30 socket_get_status() : Alias de [stream_get_meta_data](#)

Cette fonction est un alias de : [stream_get_meta_data](#) .

8.83.31 socket_set_blocking() : Alias de [stream_set_blocking](#)

Cette fonction est un alias de : [stream_set_blocking](#) .

8.83.32 socket_set_timeout() : Alias de [stream_set_timeout](#)

Cette fonction est un alias de : [stream_set_timeout](#) .

8.83.33 syslog() : Génère un message dans l'historique système

bool **syslog** (int priority , string message)

`syslog` génère un message qui sera inscrit dans l'historique par le système. `priority` est une combinaison des valeurs d'accès et de niveau, qui seront décrites dans la prochaine section. Le dernier argument est le message à envoyer. Attention : les caractères `%m` seront remplacés par l'erreur (sous forme de chaîne), présente dans `errno` .

Constante	Description
<code>LOG_EMERG</code>	système inutilisable
<code>LOG_ALERT</code>	une décision doit être prise immédiatement
<code>LOG_CRIT</code>	condition critique
<code>LOG_ERR</code>	condition d'erreur
<code>LOG_WARNING</code>	condition d'alerte
<code>LOG_NOTICE</code>	condition normale, mais significative
<code>LOG_INFO</code>	message d'information
<code>LOG_DEBUG</code>	message de débogage

Utilisation de <code>syslog</code>
<pre> <?php define_syslog_variables(); // ouverture de syslog, ajout du PID et envoi simultané du // message à la sortie standard et à un mécanisme // spécifique openlog("myScriptLog", LOG_PID LOG_PERROR, LOG_LOCAL0); // quelques lignes de code if (authorized_client()) { // faire quelquechose } else { // client non autorisé! // notation de la tentative \$access = date("Y/m/d H:i:s"); syslog(LOG_WARNING, "Client non autorisé: \$access \$_SERVER[REMOTE_ADDR] (\$_SERVER[HTTP_USER_A } closelog(); ?> </pre>

Pour plus d'informations sur comment mettre en place un gestionnaire d'historique, reportez-vous au manuel Unix, page 5 `syslog.conf`

- 5

. D'autres informations sur les systèmes d'historique et leurs options sont aussi disponibles dans le manuel `syslog`

- 3

des machines Unix.

Avec Windows NT, l'historique est pris en charge par Event Log.

Note
L'utilisation de <code>LOG_LOCAL0</code> à <code>LOG_LOCAL7</code> pour le paramètre <code>facility</code> de la fonction <code>openlog</code> n'est pas disponible sous Windows.

Voir aussi `define_syslog_variables` , `openlog` et `closelog` .

8.84 NIS

8.84.1 Introduction

NIS (feu Yellow Pages / Pages jaunes) permet la gestion par le réseau de fichiers d'administration importants (tel un fichier de mot de passe). Pour plus d'informations, reportez-vous au manuel NIS, ou à [Introduction to YP/NIS](#) Introduction to YP/NIS (en anglais). Il existe un livre en anglais [Managing NFS and NIS](#) " par Hal Stern.

Note
Cette extension a été déplacée dans le module PECL et ne sera plus intégrée dans PHP à partir de PHP 5.1.0.
Note
Cette extension n'est pas disponible sur les plates-formes Windows.

8.84.2 Pré-requis

Aucun pré-requis pour pouvoir accéder à ces fonctionnalités, en dehors des fonctions standards Unix qui sont toujours disponibles (soit libc , soit libnsl : le script de configuration détectera celle que vous utilisez).

8.84.3 Installation

Pour ajouter ces fonctionnalités, vous devez compiler PHP avec l'option `--enable-yp` .

8.84.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.84.5 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

YPERR_BADARGS ([entier](#))
 YPERR_BADDB ([entier](#))
 YPERR_BUSY ([entier](#))
 YPERR_DOMAIN ([entier](#))
 YPERR_KEY ([entier](#))
 YPERR_MAP ([entier](#))
 YPERR_NODOM ([entier](#))
 YPERR_NOMORE ([entier](#))
 YPERR_PMAP ([entier](#))
 YPERR_RESRC ([entier](#))
 YPERR_RPC ([entier](#))
 YPERR_YPBIND ([entier](#))
 YPERR_YPERR ([entier](#))
 YPERR_YPSERV ([entier](#))

YPERR_VERS (entier)

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Constantes pré-définies](#)
- [yp_all](#)
- [yp_cat](#)
- [yp_err_string](#)
- [yp_errno](#)
- [yp_first](#)
- [yp_get_default_domain](#)
- [yp_master](#)
- [yp_match](#)
- [yp_next](#)
- [yp_order](#)

8.84.7 yp_cat() : Retourne un tableau contenant la carte entière

array **yp_cat** (string domain , string map)

[yp_cat](#) retourne toutes les entrées de la carte sous forme de tableau associatif dont les clés et les valeurs sont respectivement les clés et les entrées de la carte.

8.84.8 yp_err_string() : Retourne la chaîne correspondant à l'erreur donnée

string **yp_err_string** (int errorcode)

[yp_err_string](#) retourne la chaîne d'erreur associée au code erreur errorcode . Très pratique pour savoir exactement ce qui n'a pas fonctionné.

Exemple avec [yp_err_string](#)

```
<?php
    echo "Erreur : " . yp_err_string(yp_errno());
?>
```

Voir aussi [yp_errno](#) .

8.84.9 yp_errno() : Retourne le code d'erreur de la dernière opération NIS

int **yp_errno** (void)

[yp_errno](#) retourne le code d'erreur de la dernière opération.

Les erreurs possibles sont :

- 1 Mauvais arguments

- 2 Erreur RPC - Le domaine n'existe pas
- 3 Impossible de se connecter à ce domaine
- 4 Impossible de trouver cette carte sur le serveur
- 5 Cette clé n'existe pas
- 6 Erreur yp interne ou erreur client
- 7 Erreur d'allocation de ressources
- 8 Il n'y a plus d'enregistrements dans cette carte
- 9 Impossible de communiquer avec le portmapper
- 10 Impossible de communiquer avec ypbind
- 11 Impossible de communiquer avec ypserv
- 12 Le nom de domaine local n'est pas assigné
- 13 La base de données yp n'est pas valide
- 14 Erreur de version yp
- 15 Accès interdit
- 16 Base de données occupée

Voir aussi [yp_err_string](#) .

8.84.10 [yp_first\(\)](#) : Retourne le premier couple (clé ; valeur) d'une carte donnée

array [yp_first](#) (string domain , string map)

[yp_first](#) retourne le premier couple (clé ; valeur) d'une carte donnée, ou FALSE .

Exemple avec [yp_first](#)

```
<?php
$entry = yp_first($domain, "passwd.byname");

$key = key($entry);
$value = $entry[$key];

echo "La première entrée de cette carte est " . $key
     . " et sa valeur est " . $value;
?>
```

Voir aussi [yp_next](#) et [yp_get_default_domain](#) .

8.84.11 [yp_get_default_domain\(\)](#) : Retourne le domaine NIS par défaut

string [yp_get_default_domain](#) (void)

[yp_get_default_domain](#) retourne le nom de domaine NIS par défaut. Ce nom de domaine peut être utilisé pour les futurs appels NIS.

Un domaine NIS peut être décrit comme un regroupement de cartes NIS. Tous les hôtes qui ont besoin d'informations s'attachent à un domaine. Référez-vous aux documents cités en début de document pour plus de détails.

Exemple avec le domaine par défaut

```
<?php
$domain = yp_get_default_domain();
echo 'Le domaine par défaut est : ' . $domain;
```

```
?>
```

8.84.12 `yp_master()` : Retourne le nom de la machine maître pour une carte

string `yp_master` (string domain , string map)

`yp_master` retourne le nom de la machine maître d'une carte.

Exemple de maître NIS

```
<?php
$number = yp_master($domain, $mapname);
echo 'Le maître pour cette carte est : ' . $master;
?>
```

Voir aussi [yp_get_default_domain](#) .

8.84.13 `yp_match()` : Retourne la ligne associée

string `yp_match` (string domain , string map , string key)

`yp_match` retourne la valeur associée à la clé passée en argument, pour la carte spécifiée, ou FALSE . La clé doit exister et être exacte.

Exemple avec `yp_match`

```
<?php
$entry = yp_match($domain, "passwd.byname", "joe");
echo 'La valeur trouvée est : ' . $entry;
?>
```

L'exemple ci-dessus va afficher :

```
joe:##joe:11111:100:Joe User:/home/j/joe:/usr/local/bin/bash
```

Voir aussi [yp_get_default_domain](#) .

8.84.14 `yp_next()` : Retourne le couple (clé ; valeur) suivant d'une carte donnée

array `yp_next` (string domain , string map , string key)

`yp_next` retourne le couple (clé ; valeur) suivant la clé donnée d'une carte donnée ou FALSE .

Exemple avec `yp_next`

```
<?php
$entry = yp_next($domain, "passwd.byname", "joe");
if (!$entry) {
    echo "Plus d'autres entrées.\n";
    echo "";
}
$key = key($entry);
```

```
echo "L'entrée suivante après \"joe\" a la clé " . $key
      . " et la valeur " . $entry[$key];
?>
```

Voir aussi [yp_first](#) et [yp_get_default_domain](#) .

8.84.15 [yp_order\(\)](#) : Retourne le numéro d'ordre d'une carte

int [yp_order](#) (string domain , string map)

[yp_order](#) retourne le numéro d'ordre d'une carte ou FALSE .

Exemple avec [yp_order](#)

```
<?php
$number = yp_order($domain,$mapname);
echo 'Le numéro d\'ordre de cette carte est : ' . $number;
?>
```

Voir aussi [yp_get_default_domain](#) .

8.85 Lotus Notes

8.85.1 Introduction

Attention
Ce module est EXPERIMENTAL . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez ce module à vos risques et périls.
Note
Cette extension a été déplacée dans le module <u>PECL</u> et ne sera plus intégrée dans PHP à partir de PHP 5.0.0. Des informations comme les notes sur les nouvelles versions, les téléchargements, les sources des fichiers, les informations concernant les mainteneurs ainsi qu'un CHANGELOG, peuvent être trouvées ici : http://pecl.php.net/package/notes .
Attention
Cette extension est actuellement non supportée, nous cherchons un mainteneur.

- [Introduction](#)
- [notes_body](#)
- [notes_copy_db](#)
- [notes_create_db](#)
- [notes_create_note](#)
- [notes_drop_db](#)
- [notes_find_note](#)
- [notes_header_info](#)
- [notes_list_msgs](#)
- [notes_mark_read](#)
- [notes_mark_unread](#)
- [notes_nav_create](#)
- [notes_search](#)
- [notes_unread](#)
- [notes_version](#)

8.85.3 notes_copy_db() : Copie une base de données Lotus Notes

bool **notes_copy_db** (string from_database_name , string to_database_name)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.85.4 notes_create_db() : Crée une base Lotus Notes

bool **notes_create_db** (string database_name)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.85.5 notes_create_note() : Crée une note à partir du formulaire form_name

bool **notes_create_note** (string database_name , string form_name)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.85.6 notes_drop_db() : Supprime une base Lotus Notes

bool **notes_drop_db** (string database_name)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.85.7 notes_find_note() : Retourne un identifiant de note, dans le serveur database_name

int **notes_find_note** (string database_name , string name , string type)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.85.8 notes_header_info() : Ouvre le message msg_number dans la boîte aux lettres et le serveur spécifiés

object **notes_header_info** (string server , string mailbox , int msg_number)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.85.9 notes_list_msgs() : Retourne les notes rangées dans database_name

bool notes_list_msgs (string db)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.85.10 notes_mark_read() : Marque la note note_id comme lue par l'utilisateur user_name

bool notes_mark_read (string database_name , string user_name , string note_id)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.85.11 notes_mark_unread() : Marque la note note_id comme non lue pour l'utilisateur user_name

bool notes_mark_unread (string database_name , string user_name , string note_id)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.85.8 notes_header_info() : Ouvre le message msg_number dans la boîte aux lettres et le serveur spécifiés

8.85.12 notes_nav_create() : Crée un nom de navigateur dans database_name

bool **notes_nav_create** (string database_name , string name)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.85.13 notes_search() : Cherche les notes par mots-clés dans database_name

array **notes_search** (string database_name , string keywords)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.85.14 notes_unread() : Retourne l'identifiant de la note non lue pour l'utilisateur user_name

array **notes_unread** (string database_name , string user_name)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.85.15 notes_version() : Lit la version de Lotus Notes

float **notes_version** (string database_name)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.86 Netscape Server API

8.86.1 Introduction

Ces fonctions ne sont disponibles que lorsque vous utilisez PHP comme module NSAPI avec les serveurs Netscape/iPlanet/SunONE.

8.86.2 Installation

Pour l'installation de PHP sur les serveurs Netscape/iPlanet/SunONE voyez la section NSAPI ([UNIX](#) , [Windows](#)) dans le chapitre d'installation.

8.86.3 Configuration à l'exécution

Le comportement du module PHP NSAPI est affecté par le fichier de configuration php.ini . Les configurations du fichier php.ini peuvent être remplacées par d'autres paramètres de la fonction php4_execute dans obj.conf

Nom	Par défaut	Modifiable	Historique
nsapi.read_timeout	60	PHP_INI_ALL	Disponible depuis PHP 4.3.3.

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [Directives du php.ini](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

nsapi.read_timeout entier

Configure la durée d'attente des données envoyées en méthode POST par le client.

8.86.4 Types de ressources

Cette extension ne définit aucune ressource.

8.86.5 Constantes pré-définies

Cette extension ne définit aucune constante.

8.86.6 Voir aussi

NSAPI implémente un sous-ensemble des fonctions du module Apache, pour un maximum de compatibilité.

Fonction Apache (uniquement un alias)	Fonction NSAPI	Description
apache_request_headers	nsapi_request_headers	Lit tous les en-têtes de requête HTTP
apache_response_headers	nsapi_response_headers	

		Lit tous les en-têtes de réponse HTTP
getallheaders	nsapi_request_headers	Lit tous les en-têtes de requête HTTP
virtual	nsapi_virtual	Fait une sous-requête NSAPI

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Voir aussi](#)
- [nsapi_request_headers](#)
- [nsapi_response_headers](#)
- [nsapi_virtual](#)

8.86.8 nsapi_response_headers() : Lit tous les en-têtes de réponse HTTP sur Netscape serveur

array **nsapi_response_headers** (void)

[nsapi_response_headers](#) lit tous les en-têtes de réponse HTTP sur Netscape serveur.

8.86.9 nsapi_virtual() : Effectue une sous-requête NSAPI

bool **nsapi_virtual** (string uri)

[virtual](#) est une fonction spécifique NSAPI, qui est l'équivalent de `<!--#include virtual...-->` en SSI (fichiers .shtml). Elle effectue une sous-requête NSAPI. C'est pratique pour inclure des scripts CGI ou .shtml ou d'autres choses que vous voulez faire analyser par le serveur web.

Pour exécuter une sous-requête, tous les buffers doivent être vidés vers le navigateur, ainsi que les en-têtes.

Vous ne pouvez pas faire de requêtes récursives avec cette fonction, vers un autre script PHP. Si vous voulez inclure des scripts PHP utilisez la fonction [include](#) ou [require](#) .

<p>Note</p> <p>Cette fonction dépend d'une fonctionnalité non documentée de Netscape/iPlanet/SunONE. Utilisez phpinfo pour vérifier si elle est disponible. En environnement Unix, cela devrait toujours fonctionner, sous Windows, cela dépend du nom du fichier ns-httpdXX.dll .</p> <p>Lisez la note à propos des sous-requêtes dans la section NSAPI (UNIX , Windows) si vous avez ce problème.</p>

8.87 agrégation d'objets/Composition de fonctions

Attention

Ce module est **EXPERIMENTAL** . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez ce module à vos risques et périls.

8.87.1 Introduction

En programmation objet, il est courant de rencontrer la combinaison de classes simples (et de leurs instances) en une classe plus complexe. C'est une stratégie habile pour mettre en place des objets complexes, et des hiérarchies d'objets. Ce système peut fonctionner comme une alternative dynamique à l'héritage multiple. Il y a deux solutions pour combiner deux classes, suivant la relation de leurs éléments constitutifs : L' **Association** et l' **agrégation** .

Une **Association** est une combinaison d'éléments construits indépendamment et visibles à l'extérieur. Lorsque nous associons des classes ou objets, chacun garde une référence à l'autre partie de l'association. Lorsque nous associons des classes statiquement, une classe contient une référence à une instance de l'autre classe. Par exemple :

Association de classes

```
<?php
class DateTime {

    function DateTime() {
        // constructeur vide
    }

    function now() {
        return date("Y-m-d H:i:s");
    }
}

class Report {
    var $_dt = new DateTime();
    // autres propriétés ...

    function Report() {
        // initialisation du code ...
    }

    function generateReport() {
        $dateTime = $this->_dt->now();
        // autre code
    }

    // autres méthodes ...
}

$rep = new Report();
?>
```

Nous pouvons aussi associer des instances dynamiquement, en passant une référence au constructeur (ou par une autre méthode), ce qui permet de gérer dynamiquement l'association entre les objets. Nous allons modifier l'exemple ci-dessus pour illustrer ce point :

Association d'objets

```
<?php
class DateTime {
    // identique au précédent exemple
```

```

}

class DateTimePlus {
    var $_format;

    function DateTimePlus($format="Y-m-d H:i:s") {
        $this->_format = $format;
    }

    function now() {
        return date($this->_format);
    }
}

class Report {
    var $_dt;    // Nous allons garder la référence à DateTime ici
    // autre propriété

    function Report() {
        // initialisation
    }

    function setDate($dt) {
        $this->_dt =& $dt;
    }

    function generateReport() {
        $dateTime = $this->_dt->now();
        // autre code ...
    }

    // autres méthodes ...
}

$rep = new Report();
$dt = new DateTime();
$dtp = new DateTimePlus("l, F j, Y (h:i:s a, T)");

// Génération du rapport avec une simple date
$rep->setDate($dt);
echo $rep->generateReport();

// plus loin dans le code

// generation du rapport avec une date designée
$rep->setDate($dtp);
$output = $rep->generateReport();
// sauvegarde pour affichage dans la base
// ... etc ...
?>

```

L'**agrégation**, d'un autre côté, implique l'encapsulation et le masquage des parties de la combinaison. Nous pouvons agréger des classes en utilisant une méthode statique, grâce aux sous-classes (mais PHP ne supporte pas bien les sous-classes), et, dans ce cas, la définition de la classe agrégée n'est pas accessible, sauf via les méthodes de la classe contenant. L'agrégation d'instances (agrégation d'objets) implique la création dynamique de sous-objets à l'intérieur d'un autre objet et, dans le même temps, l'extension des capacités de l'objet principal (en terme de méthodes accessibles).

L'agrégation d'objets est une méthode naturelle pour représenter des relations de type tout-partie (par exemple, une molécule est une agrégation d'atomes), ou bien peut être utilisée pour obtenir un effet équivalent à l'héritage multiple, sans avoir à lier plusieurs classes et leurs interfaces. En fait, les agrégations d'objets sont plus souples, car nous pouvons alors sélectionner les méthodes et les propriétés qui sont transmises à l'objet agrégé.

8.87.2 Exemples

Nous définissons trois classes, qui implémentent chacune une méthode de stockage différente :

storage_classes.inc

```

<?php
class FileStorage {
    var $data;

    function FileStorage($data) {
        $this->data = $data;
    }
    function write($name) {
        $fp = fopen(name, "w");
        fwrite($fp, $this->data);
        fclose($data);
    }
}

class WDDXStorage {
    var $data;
    var $version = "1.0";
    var $_id; // variable "privée"

    function WDDXStorage($data) {
        $this->data = $data;
        $this->_id = $this->_genID();
    }

    function store() {
        if ($this->_id) {
            $pid = wddx_packet_start($this->_id);
            wddx_add_vars($pid, "this->data");
            $packet = wddx_packet_end($pid);
        } else {
            $packet = wddx_serialize_value($this->data);
        }
        $dbh = dba_open("varstore", "w", "gdbm");
        dba_insert(md5(uniqid("",true)), $packet, $dbh);
        dba_close($dbh);
    }

    // méthode privée
    function _genID() {
        return md5(uniqid(rand(),true));
    }
}

class DBStorage {
    var $data;
    var $dbtype = "mysql";

    function DBStorage($data) {
        $this->data = $data;
    }

    function save() {
        $dbh = mysql_connect();
        mysql_select_db("storage", $dbh);
        $serdata = serialize($this->data);
        mysql_query("insert into vars ('$serdata',now())", $dbh);
        mysql_close($dbh);
    }
}

```

```
?>
```

Puis, nous "instancions" quelques objets issus de ces classes, et nous réalisons des agrégations et désagrégations, tout en affichant quelques résultats :

test_aggregation.php

```
<?php
include "storageclasses.inc";

// quelques utilitaires

function p_arr($arr) {
    foreach($arr as $k=>$v)
        $out[] = "\t$k => $v";
    return implode("\n", $out);
}

function object_info($obj) {
    $out[] = "Classe : ".get_class($obj);
    foreach(get_object_vars($obj) as $var=>$val)
        if (is_array($val))
            $out[] = "propriété : $var (array)\n".p_arr($val);
        else
            $out[] = "propriété : $var = $val";
    foreach(get_class_methods($obj) as $method)
        $out[] = "méthode : $method";
    return implode("\n", $out);
}

$data = array(M_PI, "kludge != cruft");

// créons quelques objets simples
$fs = new FileStorage($data);
$ws = new WDDXStorage($data);

// affichons des informations sur ces objets
echo "\$fs object\n";
echo object_info($fs).\n";
echo "\n\$ws object\n";
echo object_info($ws).\n";

// maintenant, quelques agrégations

echo "\nagrégeons \$fs avec la classe WDDXStorage\n";
aggregate($fs, "WDDXStorage");
echo "L'objet \$fs \n";
echo object_info($fs).\n";

echo "\nagrégeons le résultat avec la classe DBStorage \n";
aggregate($fs, "DBStorage");
echo "L'objet \$fs \n";
echo object_info($fs).\n";

echo "\nEt finalement, désagrégeons WDDXStorage\n";
deaggregate($fs, "WDDXStorage");
echo "L'objet \$fs \n";
echo object_info($fs).\n";

?>
```

Étudions maintenant le résultat du script pour comprendre les effets secondaires et les limitations des agrégations d'objets en PHP. D'abord, nous avons créé \$fs et \$ws et ils fournissent le bon résultat (suivant la définition de leur classe). Notez que dans le but de l'agrégation d'objets, **les**

éléments privés d'une classe ou d'un objet doivent commencer par un souligné ("_"), même s'il n'y a pas de distinction réelle entre un objet privé et un objet public.

agrégation d'objets

```
L'objet $fs
Classe : filestorage
propriété : data (array)
    0 => 3.1415926535898
    1 => kludge != cruft
méthode : filestorage
méthode : write

L'objet $ws
Classe : wddxstorage
propriété : data (array)
    0 => 3.1415926535898
    1 => kludge != cruft
propriété : version = 1.0
propriété : _id = ID::9bb2b640764d4370eb04808af8b076a5
méthode : wddxstorage
méthode : store
méthode : _genid
```

Nous agrégeons alors \$fs avec la classe WDDXStorage, et nous affichons les informations. Nous pouvons aussi voir que même si l'objet \$fs est toujours du type FileStorage, il a maintenant la propriété \$version, et la méthode store, qui sont définies dans WDDXStorage. Une chose importante à noter est que les éléments privés n'ont pas été agrégés, même s'ils sont présents dans l'objet \$ws. Un autre absent est le constructeur de WDDXStorage, qu'il n'est pas logique d'agréger.

Résultat d'agrégation

```
agrégeons \ $fs avec la classe WDDXStorage
L'objet $fs
Classe : filestorage
propriété : data (array)
    0 => 3.1415926535898
    1 => kludge != cruft
propriété : version = 1.0
méthode : filestorage
méthode : write
méthode : store
```

Le processus d'agrégation est cumulatif, ce qui fait que lorsque nous agrégeons \$fs avec la classe DBStorage, nous générons un objet qui peut utiliser n'importe laquelle des méthodes de stockage de ces classes.

Accumulation dans une agrégation

```
agrégeons le résultat avec la classe DBStorage
L'objet $fs
Classe : filestorage
propriété : data (array)
    0 => 3.1415926535898
    1 => kludge != cruft
propriété : version = 1.0
propriété : dbtype = mysql
méthode : filestorage
méthode : write
méthode : store
méthode : save
```

Finalement, de la même façon que nous avons agrégé les méthodes et propriétés dynamiquement, nous pouvons aussi les désagréger. Si nous désagrégeons la classe WDDXStorage de l'objet \$fs, nous allons obtenir :

Désagrégation d'objets

```
Et finalement, désagrégeons WDDXStorage
L'objet $fs
Classe : filestorage
propriété : data (array)
    0 => 3.1415926535898
    1 => kludge != cruft
propriété : dbtype = mysql
méthode : filestorage
méthode : write
méthode : save
```

Un point que nous n'avons pas mentionné ci-dessus et que l'agrégation ne va pas écraser les méthodes ou propriétés déjà existantes dans l'objet principal. Par exemple, la classe FileStorage définit une propriété \$data , et la classe WDDXStorage aussi. Mais cette dernière ne sera pas impliquée dans l'agrégation.

- [Introduction](#)
- [Exemples](#)
- [aggregate_info](#)
- [aggregate_methods_by_list](#)
- [aggregate_methods_by_regexp](#)
- [aggregate_methods](#)
- [aggregate_properties_by_list](#)
- [aggregate_properties_by_regexp](#)
- [aggregate_properties](#)
- [aggregate](#)
- [aggregation_info](#)
- [deaggregate](#)

8.87.4 [aggregate_methods_by_list\(\)](#) : Agrège sélectivement les méthodes d'une classe grâce à une liste

void **aggregate_methods_by_list** (object object , string class_name , array methods_list , bool exclude)

[aggregate_methods_by_list](#) agrège les méthodes de la classe class_name avec l'objet object , en filtrant les méthodes agrégées grâce à la liste methods_list , appliquée au nom de la méthode. Le paramètre optionnel exclude sert à décider si l'expression régulière sélectionne les méthodes (exclude vaut FALSE sa valeur par défaut), ou bien si elle sert à identifier les méthodes qui doivent être exclues (exclude vaut TRUE).

Le constructeur de classe et les méthodes dont le nom commence par un caractère souligné ("_"), qui sont considérées comme privées, sont toujours exclues.

Voir aussi [aggregate](#) , [aggregate_info](#) , [aggregate_methods](#) , [aggregate_methods_by_regexp](#) , [aggregate_properties](#) , [aggregate_properties_by_list](#) , [aggregate_properties_by_regexp](#) et [deaggregate](#) .

8.87.5 [aggregate_methods_by_regexp\(\)](#) : Agrège sélectivement les méthodes d'une classe grâce à une expression rationnelle

void **aggregate_methods_by_regexp** (object object , string class_name , string regexp , bool exclude)

aggregate_methods_by_regexp agrège les méthodes de la classe `class_name` avec l'objet `object`, en filtrant les méthodes agrégées grâce à l'expression rationnelle `regexp`, appliquée au nom de la méthode. Le paramètre optionnel `exclude` sert à décider si l'expression rationnelle sélectionne les méthodes (`exclude` vaut `FALSE` sa valeur par défaut), ou bien si elle sert à identifier les méthodes qui doivent être exclues (`exclude` vaut `TRUE`).

Le constructeur de classe et les méthodes dont le nom commence par un caractère souligné ("_"), qui sont considérées comme privées, sont toujours exclues.

Voir aussi aggregate, aggregate_info, aggregate_methods, aggregate_methods_by_list, aggregate_properties, aggregate_properties_by_list, aggregate_properties_by_regexp et deaggregate.

8.87.6 aggregate_methods() : Agrège dynamiquement les méthodes d'une classe à un objet

void **aggregate_methods** (object `object`, string `class_name`)

aggregate_methods agrège toutes les méthodes de la classe `class_name` avec l'objet `object`, sauf les méthodes dont le nom commence par un souligné ("_"), qui sont considérées comme privées, et donc ignorées.

Voir aussi aggregate, aggregate_info, aggregate_methods_by_list, aggregate_methods_by_regexp, aggregate_properties, aggregate_properties_by_list, aggregate_properties_by_regexp et deaggregate.

8.87.7 aggregate_properties_by_list() : Agrège sélectivement les propriétés d'une classe grâce à une liste

void **aggregate_properties_by_list** (object `object`, string `class_name`, array `properties_list`, bool `exclude`)

aggregate_properties_by_list agrège les propriétés de la classe `class_name` avec l'objet `object`, en filtrant les propriétés agrégées grâce à la liste `properties_list`, appliquée au nom de la propriété. Le paramètre optionnel `exclude` sert à décider si l'expression régulière sélectionne les propriétés (`exclude` vaut `FALSE` sa valeur par défaut), ou bien si elle sert à identifier les propriétés qui doivent être exclues (`exclude` vaut `TRUE`).

Les propriétés dont le nom commence par un caractère souligné ("_"), qui sont considérées comme privées, sont toujours exclues.

Voir aussi aggregate, aggregate_methods, aggregate_methods_by_list, aggregate_methods_by_regexp, aggregate_properties, aggregate_properties_by_regexp, aggregate_info et deaggregate.

8.87.8 aggregate_properties_by_regexp() : Agrège sélectivement les propriétés d'une classe grâce à une expression rationnelle

void **aggregate_properties_by_regexp** (object `object`, string `class_name`, string `regexp`, bool `exclude`)

8.87.5 aggregate_methods_by_regexp() : Agrège sélectivement les méthodes d'une classe grâce à une exp

[aggregate_properties_by_regexp](#) agrège les propriétés de la classe `class_name` avec l'objet `object`, en filtrant les propriétés agrégées grâce à l'expression rationnelle `regexp`, appliquée au nom de la propriété. Le paramètre optionnel `exclude` sert à décider si l'expression rationnelle sélectionne les propriétés (`exclude` vaut `FALSE` sa valeur par défaut), ou bien si elle sert à identifier les propriétés qui doivent être exclues (`exclude` vaut `TRUE`).

Les propriétés dont le nom commence par un caractère souligné ("_"), qui sont considérées comme privées, sont toujours exclues.

Voir aussi [aggregate](#), [aggregate_methods](#), [aggregate_methods_by_list](#), [aggregate_methods_by_regexp](#), [aggregate_properties](#), [aggregate_properties_by_list](#), [aggregate_info](#) et [deaggregate](#).

8.87.9 [aggregate_properties\(\)](#) : Agrège dynamiquement les propriétés d'une classe à un objet

void **aggregate_properties** (object `object`, string `class_name`)

[aggregate_properties](#) agrège toutes les propriétés de la classe `class_name` avec l'objet `object`, sauf les propriétés dont le nom commence par un souligné ("_"), qui sont considérées comme privées, et donc ignorées.

Voir aussi [aggregate](#), [aggregate_methods](#), [aggregate_methods_by_list](#), [aggregate_methods_by_regexp](#), [aggregate_properties_by_list](#), [aggregate_properties_by_regexp](#), [aggregate_info](#) et [deaggregate](#).

8.87.10 [aggregate\(\)](#) : Agrège dynamiquement des classes et objets

void **aggregate** (object `object`, string `class_name`)

[aggregate](#) agrège les méthodes et propriétés de la classe `class_name` dans l'objet `object`. Les méthodes et propriétés commençant par un souligné ("_") sont considérées comme privées et ignorées dans le processus d'agrégation. Les constructeurs sont aussi exclus de l'agrégation.

Voir aussi [aggregate_info](#), [aggregate_methods](#), [aggregate_methods_by_list](#), [aggregate_methods_by_regexp](#), [aggregate_properties](#), [aggregate_properties_by_list](#), [aggregate_properties_by_regexp](#) et [deaggregate](#).

8.87.11 [aggregation_info\(\)](#) : Alias de [aggregate_info](#)

Cette fonction est un alias de : [aggregate_info](#).

8.87.12 [deaggregate\(\)](#) : Desagrège un objet

void **deaggregate** (object `object`, string `class_name`)

[deaggregate](#) supprime les méthodes et propriétés de la classe `class_name`, qui ont été agrégées dans l'objet `object`. Si le paramètre optionnel `class_name` est passé, seules les méthodes et propriétés de cette classe sont désagrégées. Sinon, toutes les méthodes et propriétés agrégées

sont retirées.

Voir aussi [aggregate](#) , [aggregate_methods](#) , [aggregate_methods_by_list](#) , [aggregate_methods_by_regexp](#) , [aggregate_properties](#) , [aggregate_properties_by_list](#) , [aggregate_properties_by_regexp](#) et [aggregate_info](#) .

8.88 Fonctions Oracle

8.88.1 Introduction

Ces fonctions vous permettront d'accéder aux serveurs Oracle10, Oracle9, Oracle 8 et Oracle7 en utilisant l'interface Oracle Call-Interface (OCI). Elle supporte le liage de variables PHP avec des marqueurs Oracle, a un support complet des LOB, FILE et ROWID, et vous permet d'utiliser des variables définies fournies par l'utilisateur.

8.88.2 Pré-requis

Vous avez besoin des bibliothèques Oracles pour utiliser cette extension. Les utilisateur Windows doivent avoir Oracle version 8.1, et se servir de la bibliothèque php_oci8.dll .

La meilleure façon d'installer tous les fichiers nécessaires est d'utiliser le client Oracle Instant, qui est disponible depuis : <http://www.oracle.com/technology/tech/oci/instantclient/instantclient.html> . Le client Instant n'a pas besoin que les variables d'environnement ORACLE_SID ou ORACLE_HOME soient définies. Par contre, vous devez toujours définir les variables LD_LIBRARY_PATH et NLS_LANG.

Avant d'utiliser cette extension, assurez-vous que vous avez bien paramétré vos variables d'environnement Oracle, ainsi que votre démon utilisateur. Ces variables doivent être définies **avant** de démarrer votre serveur web. Les variables dont vous pouvez avoir besoin sont :

- ORACLE_HOME
- ORACLE_SID
- LD_PRELOAD
- LD_LIBRARY_PATH
- NLS_LANG
- ORA_NLS33
- NLS_LANG

Pour les variables d'environnement Oracle moins fréquemment utilisées comme TNS_ADMIN, TWO_TASK, ORA_TZFILE ainsi que les diverses configurations globales Oracle comme les variables ORA_NLS33, ORA_NLS10 et NLS_*, référez-vous à la documentation Oracle.

Après avoir configuré ces variables pour votre utilisateur "serveur web", assurez-vous aussi d'ajouter cet utilisateur (nobody , www) au groupe Oracle.

Note

Si votre serveur web ne démarre pas ou plante au démarrage avec Oracle

Vérifiez que Apache a bien été compilé avec la bibliothèque pthread :

Configuration Apache avec pthread

```
# ldd /www/apache/bin/httpd
  libpthread.so.0 => /lib/libpthread.so.0 (0x4001c000)
  libm.so.6 => /lib/libm.so.6 (0x4002f000)
  libcrypt.so.1 => /lib/libcrypt.so.1 (0x4004c000)
  libdl.so.2 => /lib/libdl.so.2 (0x4007a000)
  libc.so.6 => /lib/libc.so.6 (0x4007e000)
  /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Si la libpthread n'est pas listée, vous devez réinstaller Apache :

Reinstaller Apache avec pthread

```
# cd /usr/src/apache_1.3.xx
# make clean
# LIBS=-lpthread ./config.status
# make
# make install
```

Notez bien que sur certains systèmes comme UnixWare il faut utiliser libthread au lieu de libpthread. PHP et Apache doivent alors être configurés avec EXTRA_LIBS=-lthread.

8.88.3 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

Nom	Par défaut	Modifiable	Changelog
oci8.privileged_connect	"0"	PHP_INI_SYSTEM	Devient disponible avec la version 1.1 du module oci8
oci8.max_persistent	"-1"	PHP_INI_SYSTEM	Devient disponible avec la version 1.1 du module oci8
oci8.persistent_timeout	"-1"	PHP_INI_SYSTEM	Devient disponible avec la version 1.1 du module oci8
oci8.ping_interval	"60"	PHP_INI_SYSTEM	Devient disponible avec la version 1.1 du module oci8
oci8.statement_cache_size	"20"	PHP_INI_SYSTEM	Devient disponible avec la version 1.1 du module oci8
oci8.default_prefetch	"10"	PHP_INI_SYSTEM	Devient disponible avec la version 1.1 du module oci8
oci8.old_oci_close_semantics	"0"	PHP_INI_SYSTEM	Devient disponible avec la version 1.1 du module oci8

Voici un éclaircissement sur l'utilisation des directives de configuration.

oci8.privileged_connect boolean

Cette option active les connexions privilégiées en utilisant les créances externes (OCI_SYSDBA , OCI_SYSDBA).

oci8.max_persistent int

Le nombre maximal de connexions persistantes OCI8 par processus. Le fait de définir cette option à -1 signifie qu'il n'y a aucune limite.

oci8.persistent_timeout int

Le délai maximal (en secondes) q'un processs donné est autorisé à maintenir une connexion persistante. Le fait de définir cette option à -1 signifie que les connexions persistantes seront toujours maintenues.

oci8.ping_interval int

Le délai maximal (en secondes) à attendre avant d'envoyer un ping durant `oci_pconnect` . Lorsque définie à 0, les connexions persistantes seront pingugées à chaque réutilisation. Pour désactiver complètement les pings, définissez cette option à -1.

Note

Le fait de désactiver les pings rende les appels à `oci_pconnect` hautement rentables, mais cela empêche PHP de détecter les problèmes de connexion, comme les

problèmes de réseau, ou si le serveur Oracle a été éteint depuis la connexion de PHP. Consultez la documentation de la fonction `oci_pconnect` pour plus d'informations.

`oci8.statement_cache_size` int

Cette option active la mise en cache des requêtes, et spécifie le nombre de requêtes à mettre en cache. Pour désactiver la mise en cache des requêtes, définissez cette option à 0.

Note

Un énorme cache peut accroître les performances, mais causera un accroissement de la mémoire.

`oci8.default_prefetch` int

Cette option active la prérecupération des requêtes et définit le nombre de lignes par défaut à récupérer automatiquement après l'exécution de la requête.

Note

Une large prérecupération peut accroître les performances, mais causera un accroissement de la mémoire.

`oci8.old_oci_close_semantics` boolean

Cette option contrôle le comportement de la fonction `oci_close`. Activer cette option signifie que `oci_close` ne fera rien du tout ; la connexion ne sera pas fermée tant que la fin du script ne sera pas atteinte. Ceci est uniquement pour assurer une compatibilité ascendante. Si vous pensez que vous devez activer cette option, vous êtes **vivement encouragé** à effacer les appels à la fonction `oci_close` de votre application au lieu d'activer cette option.

8.88.4 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

`OCI_DEFAULT` (entier)

Mode d'exécution des commandes. La commande n'est pas validée automatiquement lorsqu'on utilise ce mode.

`OCI_DESCRIBE_ONLY` (entier)

Mode d'exécution des commandes. Utilisez ce mode si vous ne souhaitez pas exécuter la commande, mais obtenir des descriptions.

`OCI_COMMIT_ON_SUCCESS` (entier)

Mode d'exécution des commandes. La commande est automatiquement validée après `oci_execute`.

`OCI_EXACT_FETCH` (entier)

Mode de lecture de résultats. Utilisé lorsque les applications savent à l'avance le nombre de ligne qu'il faudra lire. Ce mode désactive la lecture anticipée d'Oracle version 8 et plus récente. Le curseur est annulé une fois que le nombre de lignes à lire a été atteint, et cela réduit les ressources consommées côté serveur.

`OCI_SYSDATE` (entier)

`SQLT_BFILEE` (entier)

Utilisé avec la fonction `oci_bind_by_name` pour relier des fichiers BFILE.

`SQLT_CFILEE` (entier)

Utilisé avec la fonction `oci_bind_by_name` pour relier des fichiers CFILE.

`OCI_B_CLOB` (entier)

Utilisé avec `oci_bind_by_name` pour relier des CLOB.

`OCI_B_BLOB` (entier)

Utilisé avec `oci_bind_by_name` pour relier des BLOB.

- OCI_B_ROWID (entier)
Utilisé avec oci_bind_by_name pour relier des ROWID.
- OCI_B_CURSOR (entier)
Utilisé avec oci_bind_by_name pour relier des curseurs, précédemment alloués avec oci_new_descriptor .
- OCI_B_NTY (entier)
Utilisé avec oci_bind_by_name pour relier des noms de types de données. Note : En PHP < 5.0, il était nommé OCI_B_SQLT_NTY .
- OCI_B_BIN (entier)
- SQLT_BFILEE (entier)
Identique à OCI_B_BFILE .
- SQLT_CFILEE (entier)
Identique à OCI_B_CFILEE .
- SQLT_CLOB (entier)
Identique à OCI_B_CLOB .
- SQLT_BLOB (entier)
Identique à OCI_B_BLOB .
- SQLT_RDD (entier)
Identique à OCI_B_ROWID .
- SQLT_NTY (entier)
Identique à OCI_B_NTY .
- OCI_FETCHSTATEMENT_BY_COLUMN (entier)
Mode par défaut de oci_fetch_all .
- OCI_FETCHSTATEMENT_BY_ROW (entier)
Mode alternatif pour oci_fetch_all .
- OCI_ASSOC (entier)
Utilisé avec oci_fetch_all et oci_fetch_array pour lire un tableau associatif.
- OCI_NUM (entier)
Utilisé avec oci_fetch_all et oci_fetch_array pour lire un tableau énuméré.
- OCI_BOTH (entier)
Utilisé avec oci_fetch_all et oci_fetch_array pour lire un tableau associatif et énuméré.
- OCI_RETURN_NULLS (entier)
Utilisé avec oci_fetch_array pour obtenir des éléments vides, si la valeur du champ est NULL .
- OCI_RETURN_LOBS (entier)
Utilisé avec oci_fetch_array pour obtenir la valeur du LOB au lieu du pointeur.
- OCI_DTYPE_FILE (entier)
Cette option indique à oci_new_descriptor d'initialiser un nouveau pointeur FILE.
- OCI_DTYPE_LOB (entier)
Cette option indique à oci_new_descriptor d'initialiser un nouveau pointeur LOB.
- OCI_DTYPE_ROWID (entier)
Cette option indique à oci_new_descriptor d'initialiser un nouveau pointeur ROWID.
- OCI_D_FILE (entier)
Identique à OCI_DTYPE_FILE .
- OCI_D_LOB (entier)
Identique à OCI_DTYPE_LOB .
- OCI_D_ROWID (entier)
Identique à OCI_DTYPE_ROWID .
- OCI_SYSOPER (entier)
Utilisé avec oci_connect pour se connecter comme SYSOPER utilisant des créances externes (oci8.privileged_connect doit être activé pour utiliser cette constante).
- OCI_SYSDBA (entier)
Utilisé avec oci_connect pour se connecter comme SYSDBA utilisant des créances

externes (`oci8.privileged_connect` doit être activé pour utiliser cette constante).

`OCI_LOB_BUFFER_FREE` (entier)

Utilisé avec `OCI-Lob->flush` pour libérer les buffers utilisés.

`OCI_TEMP_CLOB` (entier)

Utilisé avec `OCI-Lob->writeTemporary` pour indiquer explicitement que un CLOB temporaire doit être créé.

`OCI_TEMP_BLOB` (entier)

Utilisé avec `OCI-Lob->writeTemporary` pour indiquer explicitement que un BLOB temporaire doit être créé.

8.88.5 Exemples

Requête basique

```
<?php

$conn = oci_connect('hr', 'hr', 'orcl');
if (!$conn) {
    $e = oci_error();
    print htmlentities($e['message']);
    exit;
}

$query = 'SELECT * FROM DEPARTMENTS';

$stmt = oci_parse($conn, $query);
if (!$stmt) {
    $e = oci_error($conn);
    print htmlentities($e['message']);
    exit;
}

$rs = oci_execute($stmt, OCI_DEFAULT);
if (!$rs) {
    $e = oci_error($stmt);
    echo htmlentities($e['message']);
    exit;
}

print '<table border="1">';
while ($row = oci_fetch_array($stmt, OCI_RETURN_NULLS)) {
    print '<tr>';
    foreach ($row as $item) {
        print '<td>'.($item?htmlentities($item):'&nbsp;').</td>';
    }
    print '</tr>';
}
print '</table>';

oci_close($conn);
?>
```

Requête Insert avec des variables liées

```
<?php

// Avant d'exécuter, création de la table
// CREATE TABLE MYTABLE (mid NUMBER, myd VARCHAR2(20));

$conn = oci_connect('scott', 'tiger', 'orcl');

$query = 'INSERT INTO MYTABLE VALUES (:myid, :mydata)';
```

```

$stid = oci_parse($conn, $query);

$cid = 60;
$data = 'Some data';

oci_bind_by_name($stid, ':myid', $cid);
oci_bind_by_name($stid, ':mydata', $data);

$r = oci_execute($stid);

if ($r)
print "Une ligne a été insérée";

oci_close($conn);

?>

```

Insertion de donnée dans une colonne CLOB

```

<?php

// Avant l'exécution, création de la table
//      CREATE TABLE MYTABLE (mykey NUMBER, myclob CLOB);

$conn = oci_connect('scott', 'tiger', 'orcl');

$mykey = 12343; // clé arbitraire pour cette exemple;

$sql = "INSERT INTO mytable (mykey, myclob)
VALUES (:mykey, EMPTY_CLOB())
RETURNING myclob INTO :myclob";

$stid = oci_parse($conn, $sql);
$clob = oci_new_descriptor($conn, OCI_D_LOB);
oci_bind_by_name($stid, ":mykey", $mykey, 5);
oci_bind_by_name($stid, ":myclob", $clob, -1, OCI_B_CLOB);
oci_execute($stid, OCI_DEFAULT);
$clob->save("A very long string");

oci_commit($conn);

// Récupération des données CLOB

$query = 'SELECT myclob FROM mytable WHERE mykey = :mykey';

$stid = oci_parse ($conn, $query);
oci_bind_by_name($stid, ":mykey", $mykey, 5);
oci_execute($stid, OCI_DEFAULT);

print '<table border="1">';
while ($row = oci_fetch_array($stid, OCI_ASSOC)) {
    $result = $row['MYCLOB']->load();
    print '<tr><td>'.$result.'</td></tr>';
}
print '</table>';

?>

```

Vous pouvez facilement accéder aux procédures stockées, de la même façon que vous le feriez par ligne de commande :

Utilisation de procédures stockées

```

<?php

```

```
// par webmaster@remoterealty.com
$sth = oci_parse($dbh, "begin sp_newaddress( :address_id, '$firstname',
'$lastname', '$company', '$address1', '$address2', '$city', '$state',
'$postalcode', '$country', :error_code );end;" );

// Cela appelle la procédure stockée sp_newaddress, avec la variable :address_id
// pour les entrées/sorties et :error_code comme variable de sortie.
// Ensuite, vous faites les liaisons suivantes :

oci_bind_by_name($sth, ":address_id", $addr_id, 10);
oci_bind_by_name($sth, ":error_code", $errorcode, 10);
oci_execute($sth);

?>
```

8.88.6 Gestion de la connexion

L'extension oci8 fournit 3 fonctions différentes pour se connecter à Oracle. Il vous appartient d'utiliser la fonction la plus appropriée pour votre application ; les informations contenues dans cette section vont vous aider à faire le bon choix.

La connexion à un serveur Oracle est une opération raisonnablement coûteuse en terme de temps que cela prend. La fonction `oci_pconnect` utilise un cache persistant de connexion qui peut être réutilisé à travers différents scripts. Cela signifie qu'une seule connexion sera utilisée par processus PHP (ou enfant Apache).

Si votre application se connecte à Oracle en utilisant un jeu différent de crédance pour chaque utilisateur web, le cache persistant utilisé par la fonction `oci_pconnect` devient moins approprié car l'augmentation du nombre concurrent d'utilisateurs va affecté les performances de votre serveur Oracle, car il devra maintenir trop de connexions en cache. Si votre application est de ce type, il est recommandé d'optimiser votre application en utilisant les options de configuration `oci8.max_persistent` et `oci8.persistent_timeout` (elles vous donnent le contrôle sur la taille et la durée de vie du cache de connexions persistentes) ou utilisez `oci_connect` à la place.

Les fonctions `oci_connect` et `oci_pconnect` employent un cache de connexion ; si vous faites des appels multiples à `oci_connect` , en utilisant les mêmes paramètres dans un script donné, le second appel ainsi que les suivants retourneront le gestionnaire de connexion existant. Le cache utilisé par la fonction `oci_connect` est nettoyé à la fin de l'exécution du script ou lorsque vous fermez explicitement le gestionnaire de connexion. `oci_pconnect` a un comportement sensiblement identique, à la différence que le cache est maintenu séparément et est conservé entre les requêtes.

Il est important de se souvenir de cette fonctionnalité de cache, car il donne l'apparence que les deux gestionnaires ne sont pas isolés transactionnellement (ils sont en faite le même gestionnaire de connexion, ils ne sont donc absolument pas isolés). Si votre connexion a besoin de deux connexions séparées, isolées transactionnellement, vous devez utiliser la fonction `oci_new_connect`

`oci_new_connect` crée toujours une nouvelle connexion au serveur Oracle, indépendamment de l'existante d'autres connexions. Les applications web à fort trafic doivent éviter d'utiliser `oci_new_connect` , particulièrement dans les sections les plus chargées de l'application.

8.88.7 Types de données supportées par le driver

Type	Liage
SQLT_NTY	Lie une collection de types natifs depuis un objet de collection PHP comme ceux créés par oci_new_collection .
SQLT_BFILEE	Lie un descripteur natif, comme ceux créés par la fonction oci_new_descriptor .
SQLT_CFILEE	Lie un descripteur natif, comme ceux créés par la fonction oci_new_descriptor .
SQLT_CLOB	Lie un descripteur natif, comme ceux créés par la fonction oci_new_descriptor .
SQLT_BLOB	Lie un descripteur natif, comme ceux créés par la fonction oci_new_descriptor .
SQLT_RDD	Lie un descripteur natif, comme ceux créés par la fonction oci_new_descriptor .
SQLT_NUM	Convertie un paramètre PHP en un type long 'C', et le lie à cette valeur.
SQLT_RSET	Lie un gestionnaire de requête natif, comme ceux créés par la fonction oci_parse ou ceux récupérés depuis d'autres requêtes OCI.
SQLT_CHR and any other type	Convertie le paramètre PHP en un type chaîne de caractères et le lie en tant que chaîne de caractères.
Type	Liage
SQLT_RSET	Crée une ressource de requêtes oci pour représenter le curseur.
SQLT_RDD	Crée un objet ROWID.
SQLT_BLOB	Crée un objet LOB.
SQLT_CLOB	Crée un objet LOB.
SQLT_BFILE	Crée un objet LOB.
SQLT_LNG	Limite à SQLT_CHR, retourné sous la forme d'une chaîne de caractères
SQLT_LBI	Limite à SQLT_BIN, retourné sous la forme d'une chaîne de caractères
Any other type	Limite à SQLT_CHR, retourné sous la forme d'une chaîne de caractères

- [Introduction](#)
- [Pré-requis](#)
- [Configuration à l'exécution](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [Gestion de la connexion](#)
- [Types de données supportées par le driver](#)
- [oci_bind_by_name](#)
- [oci_cancel](#)
- [oci_close](#)
- [OCI-Collection->append](#)
- [OCI-Collection->assign](#)
- [OCI-Collection->assignElem](#)
- [OCI-Collection->free](#)
- [OCI-Collection->getElem](#)
- [OCI-Collection->max](#)
- [OCI-Collection->size](#)
- [OCI-Collection->trim](#)
- [oci_commit](#)
- [oci_connect](#)

- [oci_define_by_name](#)
- [oci_error](#)
- [oci_execute](#)
- [oci_fetch_all](#)
- [oci_fetch_array](#)
- [oci_fetch_assoc](#)
- [oci_fetch_object](#)
- [oci_fetch_row](#)
- [oci_fetch](#)
- [oci_field_is_null](#)
- [oci_field_name](#)
- [oci_field_precision](#)
- [oci_field_scale](#)
- [oci_field_size](#)
- [oci_field_type_raw](#)
- [oci_field_type](#)
- [oci_free_statement](#)
- [oci_internal_debug](#)
- [OCI-Lob->append](#)
- [OCI-Lob->close](#)
- [oci_lob_copy](#)
- [OCI-Lob->eof](#)
- [OCI-Lob->erase](#)
- [OCI-Lob->export](#)
- [OCI-Lob->flush](#)
- [OCI-Lob->free](#)
- [OCI-Lob->getBuffering](#)
- [OCI-Lob->import](#)
- [oci_lob_is_equal](#)
- [OCI-Lob->load](#)
- [OCI-Lob->read](#)
- [OCI-Lob->rewind](#)
- [OCI-Lob->save](#)
- [OCI-Lob->saveFile](#)
- [OCI-Lob->seek](#)
- [OCI-Lob->setBuffering](#)
- [OCI-Lob->size](#)
- [OCI-Lob->tell](#)
- [OCI-Lob->truncate](#)
- [OCI-Lob->write](#)
- [OCI-Lob->writeTemporary](#)
- [OCI-Lob->writeToFile](#)
- [oci_new_collection](#)
- [oci_new_connect](#)
- [oci_new_cursor](#)
- [oci_new_descriptor](#)
- [oci_num_fields](#)
- [oci_num_rows](#)
- [oci_parse](#)
- [oci_password_change](#)
- [oci_pconnect](#)
- [oci_result](#)
- [oci_rollback](#)
- [oci_server_version](#)

- [oci_set_prefetch](#)
- [oci_statement_type](#)
- [ocibindbyname](#)
- [ocicancel](#)
- [ocicloselob](#)
- [ocicollappend](#)
- [ocicollassign](#)
- [ocicollassignelem](#)
- [ocicollgetelem](#)
- [ocicollmax](#)
- [ocicollsize](#)
- [ocicolltrim](#)
- [ocicolumnisnull](#)
- [ocicolumnname](#)
- [ocicolumnprecision](#)
- [ocicolumnscale](#)
- [ocicolumnsize](#)
- [ocicolumntype](#)
- [ocicolumntyperaw](#)
- [ocicommit](#)
- [ocidefinebyname](#)
- [ocierror](#)
- [ociexecute](#)
- [ocifetch](#)
- [ocifetchinto](#)
- [ocifetchstatement](#)
- [ocifreecollection](#)
- [ocifreecursor](#)
- [ocifreedesc](#)
- [ocifreestatement](#)
- [ociinternaldebug](#)
- [ociloadlob](#)
- [ocilogoff](#)
- [ocilogon](#)
- [ocinewcollection](#)
- [ocinewcursor](#)
- [ocinewscriptor](#)
- [ocinlogon](#)
- [ocinumcols](#)
- [ociparse](#)
- [ociplogon](#)
- [ocireult](#)
- [ocirollback](#)
- [ocirowcount](#)
- [ocisavelob](#)
- [ocisavelobfile](#)
- [ociserverversion](#)
- [ocisetprefetch](#)
- [ocistatementtype](#)
- [ociwritelobtofile](#)
- [ociwritetemporarylob](#)

8.88.9 `oci_cancel()` : Termine la lecture de ressources Oracle

bool `oci_cancel` (resource stmt)

`oci_cancel` détruit les ressources liées au dernier résultat stmt . Si vous ne souhaitez plus lire d'informations dans ce résultat, utilisez cette fonction.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction `ocicancel` . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction `oci-cancel` , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.10 `oci_close()` : Ferme une connexion Oracle

bool `oci_close` (resource connection)

`oci_close` ferme la connexion Oracle, représentée par la ressource connection .

Note

Depuis la version 1.1, `oci_close` ferme correctement les connexion Oracle. Utilisez l'option `oci8.old_oci_close_semantics` pour restaurer l'ancien comportement de cette fonction.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction `ociclose` . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction `oci-close` , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.11 `OCI-Collection->append()` : Ajoute un élément à une collection Oracle

OCI-Collection bool `append` (mixed value)

Ajoute un élément à la fin d'une collection. Le paramètre value peut être une chaîne de caractères ou un nombre.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.12 `OCI-Collection->assign()` : Assigne une valeur à une collection depuis une autre collection Oracle

OCI-Collection bool `assign` (OCI-Collection from)

`collection->assign` assigne une valeur à une collection, à partir de la collection from . Les deux collections doivent avoir été créées avec `oci_new_collection` avant d'utiliser cette fonction.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.13 OCI-Collection->assignElem() : Assigne une valeur à un élément d'une collection Oracle

OCI-Collection bool **assignElem** (int index , mixed value)

Assigne une valeur à l'élément dont l'index est index . Le paramètre value peut être une chaîne de caractères ou bien un nombre.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.14 OCI-Collection->free() : Libère les ressources associées avec un objet de collection

OCI-Collection bool **free** ()

OCI-Collection->free libère les ressources associées à la collection.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.15 OCI-Collection->getElem() : Retourne la valeur d'un élément d'une collection Oracle

OCI-Collection mixed **getElem** (int index)

Retourne la valeur de l'élément dont l'index est index . Les index sont numérotés à partir de 1.

OCI-Collection->getElem retournera FALSE si cet élément n'existe pas; NULL , si l'élément est NULL chaîne si la colonne est de type chaîne, et nombre si c'est un champs numérique.

8.88.16 OCI-Collection->max() : Retourne le nombre maximal de valeurs d'une collection Oracle

OCI-Collection int **max** ()

OCI-Collection->max retourne le nombre maximum d'éléments de la collection. Si la valeur retournée est 0, alors la collection n'est pas limitée. OCI-Collection->max retourne FALSE en cas d'erreur.

8.88.17 OCI-Collection->size() : Retourne la taille d'une collection Oracle

OCI-Collection int **size** ()

OCI-Collection->size retourne le nombre d'éléments de la collection ou FALSE si une erreur survient.

8.88.18 OCI-Collection->trim() : Supprime les derniers éléments d'une collection Oracle

OCI-Collection bool **trim** (int num)

OCI-Collection->trim supprime les num derniers éléments d'une collection.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.19 oci_commit() : Valide les transactions Oracle en cours

bool **oci_commit** (resource connection)

oci_commit valide toutes les transactions en cours sur la connexion Oracle connection .

Cet exemple montre comment utiliser oci_commit .

Exemple avec oci_commit

```
<?php
// Connexion au serveur Oracle
$conn = oci_connect('scott', 'tiger');

// Analyse de la requête SQL
$stmt = oci_parse($conn, "
                INSERT INTO
                employees (nom, surnom)
                VALUES
                ('Maxim', 'Maletsky')
                ");

/* Exécution de la requête
   OCI_DEFAULT demande à oci_execute()
   de ne pas valider immédiatement le résultat */
oci_execute($stmt, OCI_DEFAULT);

/*
Analyse et exécution d'autres commandes ici...
*/

// Validation de la transaction
$committed = oci_commit($conn);

// Vérification de la validation : si une erreur est survenue, afficher
// le message d'erreur
if (!$committed) {
    $error = oci_error($conn);
    echo 'Validation échouée. Oracle dit : ' . $error['message'];
}
?>
```

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Les transactions sont automatiquement annulées lorsque vous fermez la connexion, ou lorsque le script se termine, un des deux arrivant le premier. Vous devez explicitement appeler la fonction oci_commit pour valider la transaction ou la fonction oci_rollback pour l'annuler.

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction `ocicommit` . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction `oci_commit` , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

Voir aussi `oci_rollback` et `oci_execute` .

8.88.20 oci_connect() : Etablit une connexion avec un serveur Oracle

resource `oci_connect` (string username , string password , string db , string charset , int session_mode)

`oci_connect` retourne une ressource de connexion Oracle, nécessaire à la plupart des appels OCI. username est le nom d'utilisateur et password est le mot de passe utilisés durant l'identification. Le troisième paramètre db , optionnel, contient le nom de l'instance locale d'Oracle, ou le nom de l'entrée dans le fichier tnsnames.ora auquel vous voulez vous connecter. Si le troisième paramètre optionnel db n'est pas spécifié, PHP va utiliser la variable d'environnement ORACLE_SID (Oracle instance) ou TWO_TASK (tnsnames.ora) pour déterminer la base à laquelle se connecter.

Le paramètre session_mode est disponible depuis la version 1.1 et accepte les valeurs suivantes : OCI_DEFAULT , OCI_SYSOPER et OCI_SYSDBA . Si OCI_SYSOPER ou OCI_SYSDBA est spécifié `oci_connect` tentera d'établir une connexion privilégiée en utilisant des crédits externes. Les connexions privilégiées sont désactivées par défaut. Pour les activer, vous devez définir `oci8.privileged_connect` à On.

Note

Si vous utilisez PHP avec le client Instant Oracle, vous pouvez utiliser la méthode de nommage facile de connexion tel que décrit ici :

http://download-west.oracle.com/docs/cd/B12037_01/network.101/b10775/naming.htm#i498306 .

Basiquement, cela signifie que vous pouvez spécifier `"/db_host[:port]/database_name"` en tant que nom de base de données. ais, si vous voulez utiliser l'ancienne méthode de nommage, vous **devez** définir soit ORACLE_HOME , soit TNS_ADMIN .

Note

Le second appel ainsi que les suivants à la fonction `oci_connect` avec les mêmes paramètres retournera le gestionnaire de connexion retourné lors du premier appel. Cela signifie que les requêtes envoyées sur un gestionnaire seront également envoyées aux autres gestionnaires, car c'est le **même** gestionnaire. Ce comportement est démontré dans l'exemple 1 ci-dessous. Si vous avez besoin de deux gestionnaires isolés transactionnellement, vous devez utiliser la fonction `oci_new_connect` à la place.

Si vous utilisez un serveur Oracle version 9.2 et suivant, vous pouvez renseigner le paramètre charset , qui peut être utilisé dans la nouvelle connexion. Si vous utilisez un serveur Oracle inférieur à la version 9.2, ce paramètre sera ignoré et la variable d'environnement NLS_LANG sera utilisé à la place.

Exemple avec oci_connect

```
<?php
echo '<pre>';
$db = '';

$c1 = oci_connect("scott", "tiger", $db);
$c2 = oci_connect("scott", "tiger", $db);

function create_table($conn)
{
```

```

$stmt = oci_parse($conn, "create table scott.hallo (test varchar2(64))");
oci_execute($stmt);
echo $conn . " created table\n\n";
}

function drop_table($conn)
{
    $stmt = oci_parse($conn, "drop table scott.hallo");
    oci_execute($stmt);
    echo $conn . " dropped table\n\n";
}

function insert_data($conn)
{
    $stmt = oci_parse($conn, "insert into scott.hallo
        values('$conn' || ' ' || to_char(sysdate,'DD-MON-YY HH24:MI:SS'))");
    oci_execute($stmt, OCI_DEFAULT);
    echo $conn . " inserted hallo\n\n";
}

function delete_data($conn)
{
    $stmt = oci_parse($conn, "delete from scott.hallo");
    oci_execute($stmt, OCI_DEFAULT);
    echo $conn . " deleted hallo\n\n";
}

function commit($conn)
{
    oci_commit($conn);
    echo $conn . " committed\n\n";
}

function rollback($conn)
{
    oci_rollback($conn);
    echo $conn . " rollback\n\n";
}

function select_data($conn)
{
    $stmt = oci_parse($conn, "select * from scott.hallo");
    oci_execute($stmt, OCI_DEFAULT);
    echo $conn."----selecting\n\n";
    while (oci_fetch($stmt)) {
        echo $conn . " [" . oci_result($stmt, "TEST") . "]\n\n";
    }
    echo $conn . "----done\n\n";
}

create_table($c1);
insert_data($c1); // Insertion d'une ligne via c1
insert_data($c2); // Insertion d'une ligne via c2

select_data($c1); // Résultat des insertions
select_data($c2);

rollback($c1); // Annulation sur c1

select_data($c1); // Les deux insertions ont été annulées
select_data($c2);

insert_data($c2); // Insertion d'une ligne via c2
commit($c2); // Validation via c2

select_data($c1); // Le résultat de c2 est retourné

```

```

delete_data($c1); // Effacement de toute la table via c1
select_data($c1); // Aucune ligne de trouvée
select_data($c2); // Aucune ligne de trouvée
commit($c1); // Validation via c1

select_data($c1); // Aucune ligne de trouvée
select_data($c2); // Aucune ligne de trouvée

drop_table($c1);
echo '</pre>';
?>

```

`oci_connect` retourne FALSE si une erreur survient.

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction `ocilogon`. Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction `oci_connect`, pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

Voir aussi `oci_pconnect`, `oci_new_connect` et `oci_close`.

8.88.21 `oci_define_by_name()` : Utilise une variable PHP pour la phase de définition, dans une commande SELECT Oracle

`bool oci_define_by_name` (resource statement , string column_name , mixed variable , int type)

`oci_define_by_name` copie les valeurs issues de la colonne `column_name` de la commande exécutée `statement`, dans la variable PHP `&variable`. Notez que les colonnes Oracle sont toutes en majuscules, tandis que dans les SELECT, vous pouvez aussi les écrire en minuscules.

`oci_define_by_name` s'attend à ce que `column_name` soit en majuscules. Si vous définissez une variable qui n'existe pas dans la commande SELECT, vous ne serez pas prévenu par une erreur.

`type` est le type de données à lire. Si vous avez besoin de définir un type de données abstrait, tel (LOB/ROWID/BFILE), vous devez lui allouer la mémoire avec `oci_new_descriptor`. Reportez-vous aussi à `oci_bind_by_name`.

Exemple avec `oci_define_by_name`

```

<?php
/* Exemple oci_define_by_name - thies at thieso dot net (980219) */

$conn = oci_connect("scott", "tiger");

$stmt = oci_parse($conn, "SELECT empno, ename FROM emp");

/* La définition doit intervenir avant oci_execute! */

oci_define_by_name($stmt, "EMPNO", $empno);
oci_define_by_name($stmt, "ENAME", $ename);

oci_execute($stmt);

while (oci_fetch($stmt)) {
    echo "empno : " . $empno . "\n";
    echo "ename : " . $ename . "\n";
}

oci_free_statement($stmt);
oci_close($conn);

```

8.88.21 `oci_define_by_name()` : Utilise une variable PHP pour la phase de définition, dans une commande

?>

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction `oci_definebyname`. Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction `oci_define_by_name`, pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.22 oci_error() : Retourne la dernière erreur Oracle

array `oci_error` (*resource source*)

`oci_error` retourne la dernière erreur trouvée. Si le paramètre `source` n'est pas fourni, la dernière erreur rencontrée est retournée. Pour la plupart des erreurs, le paramètre `source` représente une ressource de connexion. Pour les erreurs de connexion avec les fonctions `oci_connect`, `oci_new_connect` ou `oci_pconnect`, ne passez aucun paramètre. Si aucune erreur n'est trouvée, `oci_error` retourne `FALSE`. `oci_error` retourne l'erreur sous la forme d'un tableau associatif. Dans ce tableau, l'élément `code` contient le numéro d'erreur Oracle et `message` contient le message d'erreur.

Note

Depuis PHP 4.3

`offset` et `sqltext` seront aussi incluses dans le tableau retourné, pour indiquer la position de l'erreur et la requête en cause.

Exemple d'affichage d'un message d'erreur Oracle après une erreur de connexion

```
$conn = @oci_connect("scott", "tiger", "mydb");
if (!$conn) {
    $e = oci_error(); // Pour les erreurs oci_connect, aucun paramètre n'est passé
    echo htmlentities($e['message']);
}
```

Exemple d'affichage d'un message d'erreur Oracle après une erreur d'analyse

```
$stmt = @oci_parse($conn, "select ' from dual"); // Notez l'erreur de guillemet
if (!$stmt) {
    $e = oci_error($conn); // Pour les erreurs oci_parse, on passe la ressource de connexion
    echo htmlentities($e['message']);
}
```

Exemple d'affichage d'un message d'erreur Oracle après une erreur d'exécution

```
$r = oci_execute($stmt);
if (!$r) {
    $e = oci_error($stmt); // Pour les erreurs oci_execute, on passe la ressource de traitement
    echo htmlentities($e['message']);
    echo '<pre>';
    echo htmlentities($e['sqltext']);
    printf("\n%.($e['offset']+1).\"s\", \"^\");
    echo '</pre>';
}
```

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction `ocierror`. Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction `oci_error`, pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.23 `oci_execute()` : Exécute une commande SQL Oracle

bool `oci_execute` (resource stmt , int mode)

`oci_execute` exécute la commande préparée `stmt` (voir `oci_parse`). L'option `mode` vous permet de spécifier le mode d'exécution (par défaut, il est à `OCI_COMMIT_ON_SUCCESS`). Si vous ne voulez pas que la commande soit automatiquement validée, utilisez le mode `OCI_DEFAULT` .

Lors de l'utilisation du mode `OCI_DEFAULT` , vous créez une transaction. Les transactions sont automatiquement annulées lorsque vous fermez la connexion, ou lorsque le script se termine, un des deux arrivant le premier. Vous devez explicitement appeler la fonction `oci_commit` pour valider la transaction, ou la fonction `oci_rollback` pour l'annuler.

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction `ociexecute` . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction `oci_execute` , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.24 `oci_fetch_all()` : Lit toutes les lignes d'un résultat Oracle

int `oci_fetch_all` (resource statement , array output , int skip , int maxrows , int flags)

`oci_fetch_all` lit toutes les lignes du résultat `statement` dans le tableau défini `&output` . `oci_fetch_all` retourne le nombre de lignes lues, ou `FALSE` en cas d'erreur. `skip` est le nombre de lignes initiales à ignorer lors de la lecture du résultat. Par défaut, ce paramètre vaut 0, pour commencer la lecture à la première ligne. `maxrows` est le nombre de lignes maximal à lire, à partir de la `skip`-ième. S'il prend la valeur de -1, cela signifie que toutes les lignes seront lues.

Note

Cette fonction définit les champs `NULL` à la valeur PHP `NULL` .

Le paramètre `flags` est la combinaison des constantes suivantes :

- `OCI_FETCHSTATEMENT_BY_ROW`
- `OCI_FETCHSTATEMENT_BY_COLUMN` (par défaut)
- `OCI_NUM`
- `OCI_ASSOC`

Exemple avec `oci_fetch_all`

```
<?php
/* oci_fetch_all exemple mbritton at verinet dot com (990624) */

$conn = oci_connect("scott", "tiger");

$stmt = oci_parse($conn, "select * from emp");

oci_execute($stmt);

$rows = oci_fetch_all($stmt, $results);
if ($rows > 0) {
    echo "<table border='1'\>\n";
    echo "<tr>\n";
    foreach ($results as $key => $val) {
        echo "<th>$key</th>\n";
    }
}
```

```

echo "</tr>\n";

for ($i = 0; $i < $nrows; $i++) {
    echo "<tr>\n";
    foreach ($results as $data) {
        echo "<td>$data[$i]</td>\n";
    }
    echo "</tr>\n";
}
echo "</table>\n";
} else {
    echo "Pas de ligne<br />\n";
}
echo "$nrows Lignes lues<br />\n";

oci_free_statement($stmt);
oci_close($conn);
?>

```

Pour plus de détails sur le mapping des types de données affectué par le driver oci8, lisez les [types de données supportés par le driver](#)

[oci_fetch_all](#) retourne FALSE en cas d'erreur.

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction [ocifetchstatement](#) . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction [oci_fetch_all](#) , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.25 [oci_fetch_array\(\)](#) : Lit une ligne d'un résultat Oracle sous forme de tableau

array [oci_fetch_array](#) (resource statement , int mode)

[oci_fetch_array](#) retourne un tableau qui correspond à la prochaine ligne disponible dans le résultat statement , ou bien FALSE en cas d'erreur.

[oci_fetch_array](#) retourne un tableau, indexé numériquement et avec les noms des colonnes.

Note

Cette fonction définit les champs NULL à la valeur PHP NULL .

Le paramètre optionnel mode peut être la combinaison des constantes suivantes :

- OCI_BOTH - retourne un tableau, indexé numériquement et avec les noms de colonnes. Identique à OCI_ASSOC + OCI_NUM). C'est le comportement par défaut.
- OCI_ASSOC - retourne un tableau associatif. C'est le même comportement que celui de la fonction [oci_fetch_assoc](#) .
- OCI_NUM - retourne un tableau indexé numériquement, C'est le même comportement que celui de la fonction [oci_fetch_row](#) .
- OCI_RETURN_NULLS - Crée des éléments vides pour les valeurs NULL .
- OCI_RETURN_LOBS - Retourne la valeur du LOB.

La configuration par défaut de mode est OCI_BOTH .

Il faut mentionner ici que `oci_fetch_array` n'est pas **significativement** plus lente que `oci_fetch_row`, mais bien plus pratique.

Note

Oracle retourne tous les noms de champs en majuscules, et les noms des index du tableau résultant seront donc aussi en majuscules.

Exemple avec `oci_fetch_array` et `OCI_BOTH`

```
<?php
$connection = oci_connect("apelsin", "kanistra");

$query = "SELECT id, name FROM fruits";

$stmt = oci_parse ($connection, $query);
oci_execute ($stmt);

while ($row = oci_fetch_array ($stmt, OCI_BOTH)) {
    echo $row[0] . ' et ' . $row['ID'] . ' sont identiques<br />';
    echo $row[1] . ' et ' . $row['NAME'] . ' sont identiques<br />';
}
?>
```

Exemple avec `oci_fetch_array` et `OCI_NUM`

```
<?php
$connection = oci_connect("user", "password");

$query = "SELECT id, name, lob_field FROM fruits";

$stmt = oci_parse ($connection, $query);
oci_execute ($stmt);

while ($row = oci_fetch_array ($stmt, OCI_NUM)) {
    echo $row[0] . '<br />';
    echo $row[1] . '<br />';
    echo $row[2]->read(100) . '<br />'; //Affiche les 100 premiers bytes du LOB
}
?>
```

Exemple avec `oci_fetch_array` et `OCI_ASSOC`

```
<?php
$connection = oci_connect("user", "password");

$query = 'SELECT id, name, lob_field FROM fruits';

$stmt = oci_parse ($connection, $query);
oci_execute ($stmt);

while ($row = oci_fetch_array ($stmt, OCI_NUM)) {
    echo $row['ID'] . '<br />';
    echo $row['NAME'] . '<br />';
    echo $row['LOB_FIELD'] . '<br />'; //Affiche "Object id #1"
}
?>
```

Exemple avec `oci_fetch_array` et `OCI_RETURN_LOBS`

```
<?php
$connection = oci_connect("user", "password");
```

```

$query = 'SELECT id, name, lob_field FROM fruits';

$stmt = oci_parse ($connection, $query);
oci_execute ($stmt);

while ($row = oci_fetch_array ($stmt, OCI_NUM)) {
    echo $row[0] . '<br />';
    echo $row[1] . '<br />';
    echo $row['LOB_FIELD'] . '<br />'; //Affiche le contenu du LOB
}
?>

```

Pour plus de détails sur le mapping des types de données affectué par le driver oci8, lisez les [types de données supportés par le driver](#)

Voir aussi [oci_fetch_assoc](#) , [oci_fetch_object](#) , [oci_fetch_row](#) et [oci_fetch_all](#) .

8.88.26 [oci_fetch_assoc\(\)](#) : Lit une ligne d'un résultat Oracle sous forme de tableau associatif

array [oci_fetch_assoc](#) (resource statement)

[oci_fetch_assoc](#) est identique à la fonction [oci_fetch_array](#) et le mode OCI_ASSOC . Elle retourne la ligne de résultat courante, issue de la commande statement , sous la forme d'un tableau indexé avec les noms des colonnes.

Note

Cette fonction définit les champs NULL à la valeur PHP NULL .

Les prochains appels à [oci_fetch_assoc](#) retourneront la ligne suivante, et FALSE s'il n'y a plus de ligne.

Note

Oracle retourne tous les noms de champs en majuscules, et les noms des index du tableau résultant seront donc aussi en majuscules.

Pour plus de détails sur le mapping des types de données affectué par le driver oci8, lisez les [types de données supportés par le driver](#)

Voir aussi [oci_fetch_array](#) , [oci_fetch_object](#) , [oci_fetch_row](#) et [oci_fetch_all](#) .

8.88.27 [oci_fetch_object\(\)](#) : Lit une ligne d'un résultat Oracle sous forme d'objet

object [oci_fetch_object](#) (resource statement)

[oci_fetch_object](#) retourne la prochaine ligne du résultat statement sous la forme d'un objet, dont les membres correspondent aux noms des colonnes du résultat.

Note

Cette fonction définit les champs NULL à la valeur PHP NULL .

Les prochains appels à [oci_fetch_object](#) retourneront la ligne suivante, et FALSE s'il n'y a plus de lignes.

Note

Oracle retourne tous les noms de champs en majuscules, et les noms des membres de l'objet résultant seront donc aussi en majuscules.

Pour plus de détails sur le mapping des types de données affectué par le driver oci8, lisez les [types de données supportés par le driver](#)

Voir aussi [oci_fetch_array](#) , [oci_fetch_assoc](#) , [oci_fetch_row](#) et [oci_fetch_all](#) .

8.88.28 **oci_fetch_row()** : Lit une ligne d'un résultat Oracle sous forme de tableau

array **oci_fetch_row** (resource statement)

[oci_fetch_row](#) est identique à la fonction [oci_fetch_array](#) et le mode OCI_NUM . Elle retourne la ligne de résultat courante, issue de la commande statement , sous la forme d'un tableau indexé numériquement.

Note

Cette fonction définit les champs NULL à la valeur PHP NULL .

Les prochains appels à [oci_fetch_row](#) retourneront la ligne suivante, et FALSE s'il n'y a plus de lignes.

Pour plus de détails sur le mapping des types de données affectué par le driver oci8, lisez les [types de données supportés par le driver](#)

Voir aussi [oci_fetch_array](#) , [oci_fetch_object](#) , [oci_fetch_assoc](#) et [oci_fetch_all](#) .

8.88.29 **oci_fetch()** : Lit la prochaine ligne dans le résultat Oracle

bool **oci_fetch** (resource statement)

[ocifetch](#) lit la ligne de résultat courante, issue de la commande statement , et la place dans le buffer interne de résultat.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Pour plus de détails sur le mapping des types de données affectué par le driver oci8, lisez les [types de données supportés par le driver](#)

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction [ocifetch](#) . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction [oci_fetch](#) , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.30 **oci_field_is_null()** : Teste si la valeur d'une colonne Oracle est NULL

bool **oci_field_is_null** (resource stmt , mixed field)

oci_field_is_null retourne TRUE si la colonne field du résultat stmt est NULL . Vous pouvez utiliser le numéro de colonne (l'indexation des colonnes commence à 1) ou le nom de la colonne, pour le paramètre field .

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction ocicolumnisnull . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction oci_field_is_null , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.31 oci_field_name() : Retourne le nom d'une colonne Oracle

string oci_field_name (resource statement , int field)

oci_field_name retourne le nom de la colonne numéro field (l'indexation des colonnes commence à 1), dans le résultat de la commande statement .

Exemple avec oci_field_name

```
<?php
    $conn = oci_connect("scott", "tiger");
    $stmt = oci_parse($conn, "SELECT * FROM emp");
    oci_execute($stmt);

    echo "<table border=\"1\">";
    echo "<tr>";
    echo "<th>Name</th>";
    echo "<th>Type</th>";
    echo "<th>Length</th>";
    echo "</tr>";

    $ncols = oci_num_fields($stmt);

    for ($i = 1; $i <= $ncols; $i++) {
        $column_name = oci_field_name($stmt, $i);
        $column_type = oci_field_type($stmt, $i);
        $column_size = oci_field_size($stmt, $i);

        echo "<tr>";
        echo "<td>$column_name</td>";
        echo "<td>$column_type</td>";
        echo "<td>$column_size</td>";
        echo "</tr>";
    }

    echo "</table>\n";
    oci_free_statement($stmt);
    oci_close($conn);
?>
```

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction ocicolumnname . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction oci_field_name , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

Voir aussi ocinumcols , ocicolumntype et ocicolumnsize .

8.88.32 oci_field_precision() : Lit la précision d'une colonne Oracle

int **oci_field_precision** (resource statement , int field)

oci_field_precision retourne la précision du champ dont l'index est field , dans le résultat statement . Les index sont numérotés à partir de 1.

Pour les colonnes de type FLOAT, la précision et non nulle, et l'échelle est de -127. Si la précision est 0, alors la colonne est de type NUMBER. Sinon, c'est de type NUMBER(precision, scale).

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction ocicolumnprecision . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction oci_field_precision , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

Voir aussi oci_field_scale et oci_field_type .

8.88.33 oci_field_scale() : Lit l'échelle d'une colonne Oracle

int **oci_field_scale** (resource statement , int field)

oci_field_scale retourne l'échelle de la colonne d'index field , dans le résultat statement . Les index commencent à 1. La fonction retourne FALSE si la colonne n'existe pas.

Pour les colonnes de type FLOAT, la précision et non nulle, et l'échelle est de -127. Si la précision est 0, alors la colonne est de type NUMBER. Sinon, elle est de type NUMBER(precision, scale) .

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction ocicolumnscale . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction oci_field_scale , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

Voir aussi oci_field_precision et oci_field_type .

8.88.34 oci_field_size() : Retourne la taille d'un champ Oracle

int **oci_field_size** (resource stmt , mixed field)

oci_field_size retourne la taille du champ field , dans le résultat de la commande stmt . Vous pouvez utiliser l'index du champ (l'indexation commence à 1) ou le nom du champ dans le paramètre field .

Exemple avec oci_field_size

```
<?php
$conn = oci_connect("scott", "tiger");
$stmt = oci_parse($conn, "SELECT * FROM emp");
oci_execute($stmt);

echo '<table border="1">';
echo '<tr>';
echo '<th>Name</th>';
echo '<th>Type</th>';
echo '<th>Length</th>';
echo '</tr>';
```

```

$ncols = oci_num_fields($stmt);

for ($i = 1; $i <= $ncols; $i++) {
    $column_name = oci_field_name($stmt, $i);
    $column_type = oci_field_type($stmt, $i);
    $column_size = oci_field_size($stmt, $i);
    echo '<tr>';
    echo "<td>$column_name</td>";
    echo "<td>$column_type</td>";
    echo "<td>$column_size</td>";
    echo '</tr>';
}

echo '</table>';

oci_free_statement($stmt);
oci_close($conn);
?>

```

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction [ocicolumnprecision](#) . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction [oci_field_precision](#) , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

Voir aussi [oci_num_fields](#) et [oci_field_name](#) .

8.88.35 oci_field_type_raw() : Lit directement le type de colonne Oracle

int **oci_field_type_raw** (resource statement , int field)

[oci_field_type_raw](#) retourne le type brut de données Oracle, pour le champ field dans le résultat statement .

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction [ocicolumntyperaw](#) . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction [oci_field_type_raw](#) , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

Cependant, si vous souhaitez avoir le type de colonne, [oci_field_type](#) sera plus adapté.

8.88.36 oci_field_type() : Retourne le type de données d'un champ Oracle

mixed **oci_field_type** (resource stmt , int field)

[oci_field_type](#) retourne le type de données du champ correspondant au numéro du champ field dans le résultat stmt (les champs sont indexés à partir de 1).

Exemple avec [oci_field_type](#)

```

<?php
$conn = oci_connect("scott", "tiger");
$stmt = oci_parse($conn, "SELECT * FROM emp");
oci_execute($stmt);

```

```

echo '<table border="1">';
echo '<tr>';
echo '<th>Nom</th>';
echo '<th>Type</th>';
echo '<th>Longueur</th>';
echo '</tr>';

$ncols = oci_num_fields($stmt);

for ($i = 1; $i <= $ncols; $i++) {
    $column_name = oci_field_name($stmt, $i);
    $column_type = oci_field_type($stmt, $i);
    $column_size = oci_field_size($stmt, $i);

    echo '<tr>';
    echo "&<td>$column_name</td>";
    echo "<td>$column_type</td>";
    echo "<td>$column_size</td>";
    echo '</tr>';
}

echo "</table>\n";

oci_free_statement($stmt);
oci_close($conn);
?>

```

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction `ocicolumntype`. Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction `oci_field_type`, pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

Voir aussi `oci_num_fields`, `oci_field_name` et `oci_field_size`.

8.88.37 `oci_free_statement()` : Libère toutes les ressources réservées par un résultat Oracle

`bool oci_free_statement (resource statement)`

`oci_free_statement` libère toutes les ressources réservées par le résultat ou curseur Oracle `statement`. Ce dernier a été obtenu de la fonction `oci_parse`, ou directement de Oracle.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.38 `oci_internal_debug()` : Active ou désactive l'affichage des données de débogage Oracle

`void oci_internal_debug (int onoff)`

`ociinternaldebug` active ou désactive l'affichage des informations de débogage. Pour les afficher, mettez `onoff` à 1, ou sinon mettez `onoff` à 0 pour les masquer.

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction `ociinternaldebug`. Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction

oci_internal_debug , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.39 OCI-Lob->append() : Ajoute des données à un LOB Oracle

OCI-Lob bool **append** (OCI-Lob lob_from)

lob->append ajout les données lob_from à la fin du LOB.

L'écriture dans un LOB avec OCI-Lob->append échouera si la bufferisation a été préalablement activée. Vous devez désactiver la bufferisation avant d'ajouter des données. Vous aurez peut être à vider les buffers avec la fonction OCI-Lob->flush avant de la désactiver.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.40 OCI-Lob->close() : Ferme un LOB Oracle

OCI-Lob bool **close** ()

lob->close ferme un LOB Oracle. Cette fonction doit être utilisée uniquement avec OCI-Lob->writeTemporary .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.41 oci_lob_copy() : Copie un LOB Oracle

bool **oci_lob_copy** (OCI-Lob lob_to , OCI-Lob lob_from , int length)

oci_lob_copy copie length octets du LOB dans un autre LOB. Le paramètre length indique la taille des données copiées. Les anciennes données du LOB de destination seront écrasées par les nouvelles.

Si vous devez copier une partie spécifique d'un LOB vers une position particulière d'un LOB, utilisez la fonction oci_lob_seek pour déplacer le pointeur interne de LOB.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.42 OCI-Lob->eof() : Teste la fin du LOB Oracle

OCI-Lob bool **eof** ()

lob->eof retourne TRUE si le pointeur interne de LOB a atteint la fin du LOB, et FALSE sinon.

8.88.43 OCI-Lob->erase() : Ecrase une partie d'un LOB Oracle

OCI-Lob int **erase** (int offset , int length)

Écrase la partie du LOB Oracle commençant à l'offset offset , sur la longueur de length octets. Les paramètres length et offset sont optionnels. OCI-Lob->erase efface toutes les données du LOB par

8.88.38 oci_internal_debug() : Active ou désactive l'affichage des données de débogage Oracle 422

défaut.

Pour les BLOB, l'effacement signifie que la valeur existante du LOB est remplacée par le caractère 0. Pour les CLOB, ce sont des espaces.

OCI-Lob->erase retourne le nombre de caractères/octets effacés, ou bien FALSE en cas d'erreur.

8.88.44 OCI-Lob->export() : Exporte un LOB Oracle dans un fichier

OCI-Lob bool **export** (string filename , int start , int length)

OCI-Lob->export exporte le contenu du LOB dans le fichier filename . Le paramètre optionnel start indique la position de début d'exportation, et le paramètre length la quantité d'octets à exporter.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.45 OCI-Lob->flush() : Ecris les LOB Oracle sur le disque

OCI-Lob bool **flush** (int flag)

OCI-Lob->flush fait écrire les données sur le disque. Par défaut, les ressources ne sont pas libérées, mais en utilisant l'option flag avec la valeur OCI_LOB_BUFFER_FREE , vous pouvez le faire explicitement. Assurez-vous de bien savoir ce que vous faites : la prochaine lecture ou écriture dans le même LOB demandera alors une requête au serveur, et la réinitialisation des ressources. Il est recommandé d'utiliser l'option OCI_LOB_BUFFER_FREE uniquement si vous n'avez plus besoin du LOB.

OCI-Lob->flush retourne FALSE si la bufferisation n'a pas été activée, ou si une erreur est survenue.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.46 OCI-Lob->free() : Détruit un pointeur de LOB Oracle

OCI-Lob bool **free** ()

Libère les ressources associées avec le descripteur, précédemment alloué avec oci_new_descriptor .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.47 OCI-Lob->getBuffering() : Retourne l'état de bufferisation LOB d'Oracle

OCI-Lob bool **getBuffering** (void)

OCI-Lob->getBuffering retourne FALSE si la bufferisation des LOB est désactivée, et TRUE si elle l'est.

8.88.48 OCI-Lob->import() : Importe un fichier dans LOB Oracle

OCI-Lob bool **import** (string filename)

lob->import écrit les données du fichier filename dans le LOB, à partir de la position courante.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.49 oci_lob_is_equal() : Compare deux LOB/FILE Oracle

bool **oci_lob_is_equal** (OCI-Lob lob1 , OCI-Lob lob2)

oci_lob_is_equal compare les deux LOB/FILE lob1 et lob2 . oci_lob_is_equal retourne TRUE si ces objets sont égaux, et FALSE sinon.

8.88.50 OCI-Lob->load() : Returns large object's contents

OCI-Lob string **load** ()

OCI-Lob->load lit le contenu d'un LOB Oracle. Le script peut être interrompu à cause de memory_limit , si ce dernier dépasse la limite. Dans la plupart des cas, il est recommandé d'utiliser la fonction OCI-Lob->read à la place. En cas d'erreur, OCI-Lob->load retourne FALSE .

8.88.51 OCI-Lob->read() : Lit une partie d'un LOB Oracle

OCI-Lob string **read** (int length)

OCI-Lob->read lit length octets à partir de la position courante du LOB. La lecture stoppe lorsque length ont été lus, ou que la fin du LOB a été atteinte. Le pointeur de LOB sera déplacé par cette lecture.

OCI-Lob->read retourne FALSE en cas d'erreur.

8.88.52 OCI-Lob->rewind() : Ramène le pointeur interne d'un LOB Oracle au début

OCI-Lob bool **rewind** ()

OCI-Lob->rewind place le pointeur interne de LOB au début du LOB.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.53 OCI-Lob->save() : Sauve des données dans un LOB Oracle

OCI-Lob bool **save** (string data , int offset)

OCI-Lob->save sauve le contenu de la variable data dans un LOB Oracle. Le paramètre offset sert à indiquer l'offset de début d'écriture dans le LOB.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.54 OCI-Lob->saveFile() : Alias de oci_lob_import

Cette fonction est un alias de : oci_lob_import .

8.88.55 OCI-Lob->seek() : Déplace le pointeur interne d'un LOB Oracle

OCI-Lob bool **seek** (int offset , int whence)

OCI->seek déplace le pointeur de fichier du LOB. Le paramètre offset indique la distance de déplacement. Le type de déplacement est spécifié avec whence :

- OCI_SEEK_SET - déplace le pointeur à la position offset
- OCI_SEEK_CUR - ajoute offset octets à la position courante
- OCI_SEEK_END - ajoute offset octets à la fin du LOB (utilisez une valeur négative pour obtenir une position avant la fin du LOB).

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.56 OCI-Lob->setBuffering() : Active/désactive la bufferisation des LOB Oracle

OCI-Lob bool **setBuffering** (bool on_off)

OCI-Lob->setBuffering active ou désactive la bufferisation des LOB Oracle, en fonction du paramètre on_off . Des appels répétés à lob->setbuffering avec la même valeur de paramètre renverra toujours TRUE . Les valeurs de on_off sont : TRUE pour activer et FALSE pour désactiver.

Utiliser cette fonction apporte des améliorations de performances par bufferisation des petites lectures et écritures de LOB : le buffer limite les allers/retours avec le serveur. oci_lob_flush doit être utilisé pour vider les buffers une fois que vous avez fini de travailler avec le LOB.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.57 OCI-Lob->size() : Retourne la taille d'un LOB Oracle

OCI-Lob int **size** ()

lob->size retourne la taille du LOB, ou bien FALSE en cas d'erreur. Les objets vides ont une taille de 0.

8.88.58 OCI-Lob->tell() : Retourne la position courante du pointeur de LOB

OCI-Lob int **tell** ()

OCI-Lob->tell retourne la position courante du pointeur interne du LOB, ou bien FALSE si une erreur survient.

8.88.59 OCI-Lob->truncate() : Tronque un LOB Oracle

OCI-Lob bool **truncate** (*int length*)

Si *length* est fourni, OCI-Lob->truncate tronque le LOB à *length* octets. Sinon, OCI-Lob->truncate va purger complètement le LOB.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.60 OCI-Lob->write() : Ecrit des données dans un LOB Oracle

OCI-Lob int **write** (*string data* , *int length*)

OCI-Lob->write écrit les données de la variable *data* à la position courante du LOB. Si le paramètre *length* est fourni, l'écriture s'arrêtera après *length* octets, ou à la fin de la variable *data* .

OCI-Lob->write retourne le nombre d'octets écrits, ou FALSE en cas d'erreur.

8.88.61 OCI-Lob->writeTemporary() : Ecrit un LOB Oracle temporaire

OCI-Lob bool **writeTemporary** (*string data* , *int lob_type*)

OCI-Lob->writeTemporary crée un LOB et y écrit le contenu de la variable *data* .

Le paramètre *lob_type* peut prendre l'une des valeurs suivantes :

- OCI_TEMP_BLOB sert à créer un BLOB temporaire.
- OCI_TEMP_CLOB sert à créer un CLOB temporaire.

Par défaut, OCI-Lob->writeTemporary crée un CLOB.

Il faut utiliser la fonction OCI-Lob->close lorsque vous avez fini de travailler avec le LOB.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.88.62 OCI-Lob->writeToFile() : Alias de oci_lob_export

Cette fonction est un alias de : oci_lob_export .

8.88.63 oci_new_collection() : Initialise une nouvelle collection Oracle

OCI-Collection **oci_new_collection** (*resource connection* , *string tdo* , *string schema*)

oci_new_collection crée un nouvel objet de collection pour la connexion *connection* . Le paramètre *tdo* doit être un type valide, écrit en majuscules. Le troisième paramètre, optionnel,

schema doit pointer sur le schéma de base de données, où le type a été créé. `oci_new_collection` utilise le nom de l'utilisateur courant comme valeur par défaut de schema .

`oci_new_collection` retourne FALSE en cas d'erreur.

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction `ocinewcollection` . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction `oci_new_collection` , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.64 `oci_new_connect()` : Etablit une nouvelle connexion à Oracle

resource `oci_new_connect` (string username , string password , string db , string charset , int session_mode)

`oci_new_connect` établit une nouvelle connexion sur un serveur Oracle et s'y identifie.

Contrairement aux fonctions `oci_connect` et `oci_pconnect` , `oci_new_connect` ne met pas en cache les connexions et retourne toujours un gestionnaire de connexion nouvellement ouvert. Ceci est très utile si votre application a besoin d'une isolation transactionnelle entre deux jeux de requêtes.

Le troisième paramètre, optionnel, db peut contenir l'instance locale d'Oracle ou un nom disponible dans le fichier tnsnames.ora . Si le troisième paramètre n'est pas fourni, PHP va chercher la valeur dans les variables d'environnement ORACLE_SID et TWO_TASK pour déterminer le nom de l'instance locale d'Oracle, et la localisation du fichier tnsnames.ora .

Le paramètre session_mode est disponible depuis la version 1.1 et accepte les valeurs suivantes : OCI_DEFAULT , OCI_SYSOPER et OCI_SYSDBA . Si OCI_SYSOPER ou OCI_SYSDBA est spécifié, `oci_new_connect` tentera d'établir une connexion privilégiée en utilisant les créances externes. Les connexions privilégiées sont désactivées par défaut. Pour les activer, vous devez définir `oci8.privileged_connect` à On.

Note

Si vous utilisez PHP avec le client Instant d'Oracle, vous pouvez utiliser les méthodes de nommage facilitant la connexion décrites ici :

http://download-west.oracle.com/docs/cd/B12037_01/network.101/b10775/naming.htm#i498306 .

Basiquement, cela signifie que vous pouvez spécifier `"/db_host[:port]/database_name"` en tant que nom de base de données. Mais si vous voulez utiliser l'ancienne façon de nommage, vous **devez** définir soit `ORACLE_HOME` , soit `TNS_ADMIN` .

Si vous utilisez un serveur Oracle version 9.2 et suivant, vous pouvez renseigner le paramètre charset , qui peut être utilisé dans la nouvelle connexion. Si vous utilisez un serveur Oracle inférieur à la version 9.2, ce paramètre sera ignoré et la variable d'environnement NLS_LANG sera utilisé à la place.

Voici comment séparer des transactions.

Exemple avec `oci_new_connect`

```
<?php
echo '<html><pre>';
$db = '';

$c1 = oci_connect("scott", "tiger", $db);
$c2 = oci_new_connect("scott", "tiger", $db);

function create_table($conn)
```

```

{
    $stmt = oci_parse($conn, "create table scott.hallo (test
varchar2(64))");
    oci_execute($stmt);
    echo $conn . " created table\n\n";
}

function drop_table($conn)
{
    $stmt = oci_parse($conn, "drop table scott.hallo");
    oci_execute($stmt);
    echo $conn . " dropped table\n\n";
}

function insert_data($conn)
{
    $stmt = oci_parse($conn, "insert into scott.hallo
        values('$conn' || ' ' || to_char(sysdate,'DD-MON-YY HH24:MI:SS'))");
    oci_execute($stmt, OCI_DEFAULT);
    echo $conn . " inserted hallo\n\n";
}

function delete_data($conn)
{
    $stmt = oci_parse($conn, "delete from scott.hallo");
    oci_execute($stmt, OCI_DEFAULT);
    echo $conn . " deleted hallo\n\n";
}

function commit($conn)
{
    oci_commit($conn);
    echo $conn . " committed\n\n";
}

function rollback($conn)
{
    oci_rollback($conn);
    echo $conn . " rollback\n\n";
}

function select_data($conn)
{
    $stmt = oci_parse($conn, "select * from scott.hallo");
    oci_execute($stmt, OCI_DEFAULT);
    echo $conn . "----selecting\n\n";
    while (oci_fetch($stmt)) {
        echo $conn . "<" . oci_result($stmt, "TEST") . ">\n\n";
    }
    echo $conn . "----done\n\n";
}

create_table($c1);
insert_data($c1);

select_data($c1);
select_data($c2);

rollback($c1);

select_data($c1);
select_data($c2);

insert_data($c2);
commit($c2);

select_data($c1);

```

```

delete_data($c1);
select_data($c1);
select_data($c2);
commit($c1);

select_data($c1);
select_data($c2);

drop_table($c1);
echo '</pre></html>';
?>

```

oci_new_connect retourne FALSE en cas d'erreur.

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction ocinlogon . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction oci_new_connect , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

Voir aussi oci_connect et oci_pconnect .

8.88.65 oci_new_cursor() : Alloue un nouveau curseur Oracle

resource **oci_new_cursor** (resource connection)

ocinewcursor alloue un nouveau pointeur de commande, pour la connexion connection .

Utiliser un REF CURSOR issu d'une procédure enregistrée

```

<?php
// supposons que votre procédure stockée info.output retourne un curseur de référence dans :data

$conn = oci_connect("scott", "tiger");
$curs = oci_new_cursor($conn);
$stmt = oci_parse($conn, "begin info.output(:data); end;");

oci_bind_by_name($stmt, "data", $curs, -1, OCI_B_CURSOR);
oci_execute($stmt);
oci_execute($curs);

while ($data = oci_fetch_row($curs)) {
    var_dump($data);
}

oci_free_statement($stmt);
oci_free_statement($curs);
oci_close($conn);
?>

```

Utiliser un REF CURSOR issu d'une commande SELECT

```

<?php
echo '<html><body>';
$conn = OCILogon("scott","tiger");
$count_cursor = "CURSOR(select count(empno) num_emps from emp " .
                "where emp.deptno = dept.deptno) as EMPCNT from dept";
$stmt = OCIParse($conn,"select deptno,dname,$count_cursor");

ociexecute($stmt);
echo '<table border="1">';
echo '<tr>';

```

```

echo '<th>DEPT NAME</th>';
echo '<th>DEPT #</th>';
echo '<th># EMPLOYEES</th>';
echo '</tr>';

while (OCIFetchInto($stmt, &$data, OCI_ASSOC)) {
    echo '<tr>';
    $dname = $data["DNAME"];
    $deptno = $data["DEPTNO"];
    echo "<td>$dname</td>";
    echo "<td>$deptno</td>";
    ociexecute($data["EMPCNT"]);
    while (OCIFetchInto($data["EMPCNT"], &$subdata, OCI_ASSOC)) {
        $num_emps = $subdata["NUM_EMPS"];
        echo "<td>$num_emps</td>";
    }
    echo '</tr>';
}
echo '</table>';
echo '</body></html>';
OCIFreeStatement($stmt);
OCILogoff($conn);
?>

```

`oci_new_cursor` retourne FALSE en cas d'erreur.

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction `ocinewcursor`. Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction `oci_new_collection`, pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.66 `oci_new_descriptor()` : Initialise un nouveau pointeur vide de LOB/FILE Oracle

OCI-Lob `oci_new_descriptor` (resource connection, int type)

`oci_new_descriptor` alloue l'espace nécessaire pour stocker un descripteur, ou un pointeur de LOB, pour la connexion connection. Les valeurs acceptées pour type sont : OCI_D_FILE, OCI_D_LOB et OCI_D_ROWID

Exemple avec `oci_new_descriptor`

```

<?php
/* Ce script est fait pour être appelé dans un formulaire HTML
 * Il attends les variables $user, $password, $table, $where, et $commitsize
 * Le script efface alors les lignes sélectionnées avec ROWID et valide
 * l'effacement après chaque groupe de $commitsize lignes.
 * (Utilisez avec prudence, car il n'y a pas d'annulation possible).
 */
$conn = oci_connect($user, $password);
$stmt = oci_parse($conn, "select rowid from $table $where");
$rowid = oci_new_descriptor($conn, OCI_D_ROWID);
oci_define_by_name($stmt, "ROWID", $rowid);
oci_execute($stmt);
while (oci_fetch($stmt)) {
    $nrows = oci_num_rows($stmt);
    $delete = oci_parse($conn, "delete from $table where ROWID = :rid");
    oci_bind_by_name($delete, ":rid", $rowid, -1, OCI_B_ROWID);
    oci_execute($delete);
    echo "$nrows\n";
    if (($nrows % $commitsize) == 0) {

```

```

        oci_commit($conn);
    }
}
$nrows = oci_num_rows($stmt);
echo "$nrows deleted...\n";
oci_free_statement($stmt);
oci_close($conn);
?>

<?php
/* Ce script illustre le chargement de LOB
 * Le formulaire utilisé dans cet exemple ressemble à ceci :
 * <form action="upload.php" method="post" enctype="multipart/form-data">
 * <input type="file" name="lob_upload" />
 * ...
 */
if (!isset($lob_upload) || $lob_upload == 'none'){
?>
<form action="upload.php" method="post" enctype="multipart/form-data">
Fichier téléchargé : <input type="file" name="lob_upload" /><br />
<input type="submit" value="Upload" /> - <input type="reset" value="Reset" />
</form>
<?php
    } else {

        // $lob_upload contient le fichier temporaire

        // Reportez-vous à la section sur le téléchargement de fichiers
        // pour sécuriser vos téléchargements

        $conn = oci_connect($user, $password);
        $lob = oci_new_descriptor($conn, OCI_D_LOB);
        $stmt = oci_parse($conn, "insert into $table (id, the_blob)
            values(my_seq.NEXTVAL, EMPTY_BLOB()) returning the_blob into :the_blob");
        oci_bind_by_name($stmt, ':the_blob', $lob, -1, OCI_B_BLOB);
        oci_execute($stmt, OCI_DEFAULT);
        if ($lob->savefile($lob_upload)){
            oci_commit($conn);
            echo "BLOB chargé !\n";
        }else{
            echo "Impossible de charger le BLOB\n";
        }
        oci_free_descriptor($lob);
        oci_free_statement($stmt);
        oci_close($conn);
    }
?>

```

Exemple avec `oci_new_descriptor`

```

<?php
/* Appel d'une procédure PL/SQL stockée qui prend un clobs
 * en entrée (PHP 4 >= 4.0.6).
 * Exemple de signature de procédure stockée PL/SQL :
 *
 * PROCEDURE save_data
 *   Argument Name                Type                In/Out Default?
 * -----
 * KEY                            NUMBER(38)         IN
 * DATA                          CLOB              IN
 *
 */

$conn = oci_connect($user, $password);
$stmt = oci_parse($conn, "begin save_data(:key, :data); end;");
$clob = oci_new_descriptor($conn, OCI_D_LOB);

```

```
oci_bind_by_name($stmt, ':key', $key);
oci_bind_by_name($stmt, ':data', $clob, -1, OCI_B_CLOB);
$clob->write($data);
oci_execute($stmt, OCI_DEFAULT);
oci_commit($conn);
$clob->free();
oci_free_statement($stmt);
?>
```

oci_new_descriptor retourne FALSE en cas d'erreur.

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction ocinewdescriptor . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction oci_new_descriptor , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.67 oci_num_fields() : Retourne le nombre de colonnes dans un résultat Oracle

int **oci_num_fields** (resource statement)

oci_num_fields retourne le nombre de colonnes dans le résultat Oracle statement .

Exemple avec oci_num_fields

```
<?php
echo "<pre>\n";
$conn = oci_connect("scott", "tiger");
$stmt = oci_parse($conn, "select * from emp");

oci_execute($stmt);

while (oci_fetch($stmt)) {
    echo "\n";
    $ncols = oci_num_fields($stmt);
    for ($i = 1; $i <= $ncols; $i++) {
        $column_name = oci_field_name($stmt, $i);
        $column_value = oci_result($stmt, $i);
        echo $column_name . ': ' . $column_value . "\n";
    }
    echo "\n";
}

oci_free_statement($stmt);
oci_close($conn);

echo "</pre>";
?>
```

oci_num_fields retourne FALSE en cas d'erreur.

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction ocinumcols . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction oci_num_fields , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.68 oci_num_rows() : Retourne le nombre de lignes affectées durant la dernière commande Oracle

int **oci_num_rows** (resource stmt)

oci_num_rows retourne le nombre de lignes affectées durant la dernière commande Oracle stmt .

Note

Cette fonction **ne retourne pas** le nombre de lignes sélectionnées. Pour les commandes de type SELECT, cette fonction va retourner le nombre de ligne qui ont été lues dans le buffer avec oci_fetch* .

Exemple avec oci_num_rows

```
<?php
    echo "<pre>";
    $conn = oci_connect("scott", "tiger");

    $stmt = oci_parse($conn, "create table emp2 as select * from emp");
    oci_execute($stmt);
    echo oci_num_rows($stmt) . " rows inserted.<br />";
    oci_free_statement($stmt);

    $stmt = oci_parse($conn, "delete from emp2");
    oci_execute($stmt, OCI_DEFAULT);
    echo oci_num_rows($stmt) . " rows deleted.<br />";
    oci_commit($conn);
    oci_free_statement($stmt);

    $stmt = oci_parse($conn, "drop table emp2");
    oci_execute($stmt);
    oci_free_statement($stmt);

    oci_close($conn);
    echo "</pre>";
?>
```

oci_num_rows retourne FALSE en cas d'erreur.

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction ocirowcount . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction oci_num_rows , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.69 oci_parse() : Prépare une requête SQL avec Oracle

resource **oci_parse** (resource connection , string query)

oci_parse analyse la requête query sur la connexion connection , et retourne TRUE si la requête query est valide, et FALSE , si ce n'est pas le cas. query peut être n'importe quelle requête SQL. oci_parse retourne une ressource de commande, qui peut être utilisée avec des fonctions comme oci_bind_by_name , oci_execute et d'autres fonctions.

Note

Cette fonction **ne valide pas** la requête query . La seule façon de savoir si la requête query est valide est de l'exécuter.

`oci_parse` retourne FALSE en cas d'erreur.

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction `ociparse`. Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction `oci_parse`, pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.70 `oci_password_change()` : Modifie le mot de passe d'un utilisateur Oracle

bool `oci_password_change` (resource connection , string username , string old_password , string new_password) resource `oci_password_change` (string dbname , string username , string old_password , string new_password)

`oci_password_change` modifie le mot de passe de l'utilisateur `username`, avec la connexion `connection` au serveur Oracle. Les paramètres `old_password` et `new_password` doivent contenir respectivement l'ancien et le nouveau mot de passe.

Note

La seconde syntaxe de `oci_password_change` est disponible depuis la version 1.1. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction `ocipasswordchange`. Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction `oci_password_change`, pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.71 `oci_pconnect()` : Ouvre une connexion persistante à un serveur Oracle

resource `oci_pconnect` (string username , string password , string db , string charset , int session_mode)

`oci_pconnect` crée une nouvelle connexion persistante à un serveur Oracle, et s'identifie avec le nom d'utilisateur `username` et le mot de passe `password`. Les connexions persistantes sont mises en cache et sont réutilisées entre les requêtes, réduisant ainsi la charge à chaque chargement de la page ; une application PHP typique a une seule connexion persistante à un serveur Oracle par processus enfant Apache (ou processus PHP FastCGI/CGI). Voir la section sur les [connexions persistantes aux bases de données](#) pour plus d'informations.

Note

Depuis la version 1.1 de l'extension `oci8`, la durée de vie ainsi que le nombre maximal de connexions persistantes Oracle peuvent être affinés en définissant les valeurs de configuration suivantes : `oci8.persistent_timeout`, `oci8.ping_interval` et `oci8.max_persistent`.

Le troisième paramètre, optionnel, `db` peut contenir l'instance locale d'Oracle ou un nom disponible dans le fichier `tnsnames.ora`. Si le troisième paramètre n'est pas fourni, PHP va chercher la valeur dans les variables d'environnement `ORACLE_SID` et `TWO_TASK` pour déterminer le nom de l'instance locale d'Oracle, et la localisation du fichier `tnsnames.ora`.

Le paramètre `session_mode` est disponible depuis la version 1.1 et accepte les valeurs suivantes : `OCI_DEFAULT`, `OCI_SYSOPER` et `OCI_SYSDBA`. Si `OCI_SYSOPER` ou `OCI_SYSDBA` sont

spécifiés, `oci_connect` tentera d'établir une connexion privilégiée en utilisant les crédances externes. Les connexions privilégiées sont désactivées par défaut. Pour les activer, vous devez définir `oci8.privileged_connect` à On.

Note

Si vous utilisez PHP avec le client Oracle Instant, vous pouvez utiliser les méthodes de nommage pour faciliter la connexion tel que décrites ici : http://download-west.oracle.com/docs/cd/B12037_01/network.101/b10775/naming.htm#i498306 . Basiquement, cela signifie que vous pouvez spécifier `"/db_host[:port]/database_name"` en tant que nom de base de données. Mais si vous voulez utiliser l'ancienne méthode de nommage, vous **devez** définir soit `ORACLE_HOME` , soit `TNS_ADMIN` .

Si vous utilisez un serveur Oracle version 9.2 et suivant, vous pouvez renseigner le paramètre `charset` , qui peut être utilisé dans la nouvelle connexion. Si vous utilisez un serveur Oracle inférieur à la version 9.2, ce paramètre sera ignoré et la variable d'environnement `NLS_LANG` sera utilisé à la place.

`oci_pconnect` retourne une ressource de connexion Oracle en cas de succès, et sinon, `FALSE` .

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction `ociplogon` . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction `oci_pconnect` , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

Voir aussi `oci_connect` et `oci_new_connect` .

8.88.72 `oci_result()` : Retourne la valeur d'une colonne dans un résultat Oracle

mixed `oci_result` (resource statement , mixed field)

`oci_result` retourne les données de la colonne `field` dans la ligne courante du résultat `statement` (voir `ocifetch`) . `ocifetch` retournera tout les types, sauf les types abstraits (ROWIDs, LOBs et FILES). `oci_result` retourne `FALSE` en cas d'erreur.

Vous pouvez utiliser le numéro de colonne (commençant à un) ou le nom de la colonne (en majuscules), dans le paramètre `field` .

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction `ociresult` . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction `oci_result` , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

Pour plus de détails sur le mapping des types de données affectué par le driver oci8, lisez les [types de données supportés par le driver](#)

Voir aussi `oci_fetch_array` , `oci_fetch_assoc` , `oci_fetch_object` , `oci_fetch_row` et `oci_fetch_all` .

8.88.73 `oci_rollback()` : Annule les transactions Oracle en cours

bool `oci_rollback` (resource connection)

`oci_rollback` annule les transactions en cours sur la connexion Oracle `connection` .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note
Les transactions sont automatiquement annulées lorsque vous fermez la connexion, ou lorsque le script se termine, un des deux arrivant le premier. Vous devez explicitement appeler la fonction oci_commit pour valider la connexion, ou la fonction oci_rollback pour l'annuler.
Note
Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction ocirollback . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction oci_rollback , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

Voir aussi [oci_commit](#) .

8.88.74 [oci_server_version\(\)](#) : Retourne une chaîne contenant les informations de version du serveur Oracle

string [oci_server_version](#) (resource connection)

[ociserverversion](#) retourne une chaîne contenant les informations de version du serveur Oracle, identifié par la connexion connection .

Exemple avec oci_server_version
<pre><?php \$conn = oci_connect("scott", "tiger"); echo "Server Version: " . oci_server_version(\$conn); oci_close(\$conn); ?></pre>
Note
Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction ociserverversion . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction oci_server_version , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.75 [oci_set_prefetch\(\)](#) : Indique le nombre de lignes qui doivent être lues à l'avance par Oracle

bool [oci_set_prefetch](#) (resource statement , int rows)

[ocisetprefetch](#) spécifie le nombre rows des premières lignes qui doivent être lues à l'avance pour la connexion Oracle statement . La valeur par défaut de rows est 1 ligne.

Note
Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction ocisetprefetch . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction oci_set_prefetch , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi l'option de configuration [oci8_default_prefetch](#) .

8.88.76 oci_statement_type() : Retourne le type de commande Oracle

string **oci_statement_type** (resource statement)

oci_statement_type retourne le type de la commande statement . oci_statement_type retourne une des valeurs suivantes :

1. SELECT
2. UPDATE
3. DELETE
4. INSERT
5. CREATE
6. DROP
7. ALTER
8. BEGIN
9. DECLARE
10. UNKNOWN

Le paramètre statement est une requête Oracle valide, retournée par oci_parse .

Exemple avec ocistatementtype

```
<?php
$conn = oci_connect("scott", "tiger");
$sql = "delete from emp where deptno = 10";

$stmt = oci_parse($conn, $sql);
if (oci_statement_type($stmt) == "DELETE") {
    die ("Nous n'êtes pas autorisés à effacer des lignes dans cette<br />");
}

oci_close($conn);
?>
```

oci_statement_type retourne FALSE en cas d'erreur.

Note

Dans les versions de PHP antérieures à la version 5.0.0, vous devez utiliser la fonction ocistatementtype . Cet ancien nom est toujours utilisable : un alias a été fait vers la fonction oci_statement_type , pour assurer la compatibilité ascendante. Toutefois, il est recommandé de ne plus l'utiliser.

8.88.77 ocibindbyname() : Alias de oci_bind_by_name

Cette fonction est un alias de : oci_bind_by_name .

8.88.78 ocicancel() : Alias de oci_cancel

Cette fonction est un alias de : oci_cancel .

8.88.79 ocicloselob() : Alias de OCI-Lob->close

Cette fonction est un alias de : OCI-Lob->close .

8.88.80 ocicollappend() : Alias de OCI-Collection->append

Cette fonction est un alias de : OCI-Collection->append .

8.88.81 ocicollassign() : Alias de OCI-Collection->assign

Cette fonction est un alias de : OCI-Collection->assign .

8.88.82 ocicollassignelem() : Alias de OCI-Collection->assignElem

Cette fonction est un alias de : OCI-Collection->assignElem .

8.88.83 ocicollgetelem() : Alias de OCI-Collection->getElem

Cette fonction est un alias de : OCI-Collection->getElem .

8.88.84 ocicollmax() : Alias de OCI-Collection->max

Cette fonction est un alias de : OCI-Collection->max .

8.88.85 ocicollsize() : Alias de OCI-Collection->size

Cette fonction est un alias de : OCI-Collection->size .

8.88.86 ocicolltrim() : Alias de OCI-Collection->trim

Cette fonction est un alias de : OCI-Collection->trim .

8.88.87 ocicolumnisnull() : Alias de oci_field_is_null

Cette fonction est un alias de : oci_field_is_null .

8.88.88 ocicolumnname() : Alias de oci_field_name

Cette fonction est un alias de : oci_field_name .

8.88.89 ocicolumnprecision() : Alias de oci_field_precision

Cette fonction est un alias de : oci_field_precision .

8.88.90 ocicolumnscale() : Alias de oci_field_scale

Cette fonction est un alias de : oci_field_scale .

8.88.91 ocicolumnsize() : Alias de oci_field_size

Cette fonction est un alias de : oci_field_size .

8.88.92 ocicolumntype() : Alias de oci_field_type

Cette fonction est un alias de : oci_field_type .

8.88.93 ocicolumntyperaw() : Alias de oci_field_type_raw

Cette fonction est un alias de : oci_field_type_raw .

8.88.94 ocicommit() : Alias de oci_commit

Cette fonction est un alias de : oci_commit .

8.88.95 ocidefinebyname() : Alias de oci_define_by_name

Cette fonction est un alias de : oci_define_by_name .

8.88.96 ocierror() : Alias de oci_error

Cette fonction est un alias de : oci_error .

8.88.97 ociexecute() : Alias de oci_execute

Cette fonction est un alias de : oci_execute .

8.88.98 ocifetch() : Alias de oci_fetch

Cette fonction est un alias de : oci_fetch .

8.88.99 ocifetchinto() : Retourne la ligne suivante d'un résultat Oracle dans un tableau

int **ocifetchinto** (resource statement , array result , int mode)

Note

Cette fonction est dépréciée. Les alternatives recommandées sont : [oci_fetch_array](#) , [oci_fetch_object](#) , [oci_fetch_assoc](#) et [oci_fetch_row](#) .

[ocifetchinto](#) retourne la ligne suivante de la commande statement (pour une commande SELECT) dans le tableau result . [ocifetchinto](#) écrasera le contenu de result . Par défaut, result sera un tableau à index numérique, commençant à 1, et qui contiendra toute les colonnes qui ne sont pas NULL.

L'option mode vous permet de modifier le comportement par défaut de la fonction. Vous pouvez passer plusieurs modes simplement en les additionnant (i.e. OCI_ASSOC+OCI_RETURN_NULLS). Les modes valides sont :

- OCI_ASSOC retourne un tableau associatif.
- OCI_NUM retourne un tableau à index numérique (DEFAULT, valeur par défaut)
- OCI_RETURN_NULLS retourne les colonnes vides.
- OCI_RETURN_LOBS retourne la valeur des objets LOB plutôt que leur descripteur.

Exemple avec [ocifetchinto](#)

```
<?php
$conn = oci_logon("username","password");

$query = "SELECT pommes FROM arbre";

$stmt = OCI_Parse ($conn, $query);
OCI_Execute ($stmt);

while (OCI_FetchInto ($stmt, $row, OCI_ASSOC)) {
    print $row['pommes'];
}
?>
```

Voir aussi [oci_fetch_array](#) , [oci_fetch_object](#) , [oci_fetch_assoc](#) , [oci_fetch_row](#) , [oci_fetch](#) et [oci_execute](#) .

8.88.100 [ocifetchstatement\(\)](#) : Alias de [oci_fetch_all](#)

Cette fonction est un alias de : [oci_fetch_all](#) .

8.88.101 [ocifreecollection\(\)](#) : Alias de [OCI-Collection->free](#)

Cette fonction est un alias de : [OCI-Collection->free](#) .

8.88.102 [ocifreecursor\(\)](#) : Alias de [oci_free_statement](#)

Cette fonction est un alias de : [oci_free_statement](#) .

8.88.103 [ocifreedesc\(\)](#) : Alias de [OCI-Lob->free](#)

Cette fonction est un alias de : [OCI-Lob->free](#) .

8.88.104 ocifreestatement() : Alias de oci_free_statement

Cette fonction est un alias de : oci_free_statement .

8.88.105 ociinternaldebug() : Alias de oci_internal_debug

Cette fonction est un alias de : oci_internal_debug .

8.88.106 ociloadlob() : Alias de OCI-Lob->load

Cette fonction est un alias de : OCI-Lob->load .

8.88.107 ocilogoff() : Alias de oci_close

Cette fonction est un alias de : oci_close .

8.88.108 ocilogon() : Alias de oci_connect

Cette fonction est un alias de : oci_connect .

8.88.109 ocinewcollection() : Alias de oci_new_collection

Cette fonction est un alias de : oci_new_collection .

8.88.110 ocinewcursor() : Alias de oci_new_cursor

Cette fonction est un alias de : oci_new_cursor .

8.88.111 ocinewscriptor() : Alias de oci_new_descriptor

Cette fonction est un alias de : oci_new_descriptor .

8.88.112 ocinlogon() : Alias de oci_new_connect

Cette fonction est un alias de : oci_new_connect .

8.88.113 ocinumcols() : Alias de oci_num_fields

Cette fonction est un alias de : oci_num_fields .

8.88.114 ociparse() : Alias de oci_parse

Cette fonction est un alias de : oci_parse .

8.88.115 ociplogon() : Alias de oci_pconnect

Cette fonction est un alias de : oci_pconnect .

8.88.116 ociresult() : Alias de oci_result

Cette fonction est un alias de : oci_result .

8.88.117 ocirollback() : Alias de oci_rollback

Cette fonction est un alias de : oci_rollback .

8.88.118 ocirowcount() : Alias de oci_num_rows

Cette fonction est un alias de : oci_num_rows .

8.88.119 ocisavelob() : Alias de OCI-Lob->save

Cette fonction est un alias de : OCI-Lob->save .

8.88.120 ocisavelobfile() : Alias de OCI-Lob->import

Cette fonction est un alias de : OCI-Lob->import .

8.88.121 ociserverversion() : Alias de oci_server_version

Cette fonction est un alias de : oci_server_version .

8.88.122 ocisetprefetch() : Alias de oci_set_prefetch

Cette fonction est un alias de : oci_set_prefetch .

8.88.123 ocistatementtype() : Alias de oci_statement_type

Cette fonction est un alias de : oci_statement_type .

8.88.124 ociwritelobtofile() : Alias de OCI-Lob->export

Cette fonction est un alias de : OCI-Lob->export .

8.88.125 ociwritetemporarylob() : Alias de OCI-Lob->writeTemporary

Cette fonction est un alias de : OCI-Lob->writeTemporary .

8.89 oggvorbis

8.89.1 Introduction

Le format de fichier OGG/Vorbis, comme définit par <http://www.vorbis.com/> et <http://www.vorbis.com/>, est un schéma pour la compression de flux audio par de multiples facteurs avec un minimum de perte de qualité. Cette extension ajoute le support Ogg Vorbis aux [gestionnaires d'URL](#) de PHP. Lorsqu'utilisé en mode lecture, les données compressées OGG/Vorbis sont déployées en audio PCM brute en un des six formats d'encodage PCM listés ci-dessous.

8.89.2 Pré-requis

Cette extension nécessite PHP \geq 4.3.0, [libogg](#) \geq 1.0, et [libvorbis](#) \geq 1.0.

8.89.3 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.89.4 Installation

oggvorbis est installé en utilisant le processus d'installation des paquets PECL.

- \$ pear install oggvorbis
- Copiez le fichier résultant oggvorbis.so dans un dossier approprié et ajoutez extension=oggvorbis.so à votre php.ini ou chargez-le dynamiquement dans votre script PHP en utilisant dl('oggvorbis.so');

8.89.5 Options de contexte

Option	Définition	Pertinence	Défaut
pcm_mode	Encodage PCM utilisé. Voir les constantes ci-dessous.	Lecture / Écriture	OGGVORBIS_PCM_S16_LE
rate	Taux d'échantillonnage PCM. Mesuré en Hz.	Écriture uniquement	44100
bitrate	Débit d'encodage moyen Vorbis / Débit d'encodage variable. Mesuré en bps (ABR) ou en niveau de qualité (VBR : 0.0 à 1.0). 128000 ABR équivaut à 0.4 VBR.	Écriture uniquement	128000
channels	Nombre de canaux PCM. 1 == Mono, 2 == Stéréo.	Écriture uniquement	2
serialno	Nombre de séries de flux dans un fichier. Doit être unique dans un fichier. Parcequ'il est potentiellement	Écriture uniquement	Random

	possible de sélectionner plusieurs nombres de séries dans un fichier chaîné, faite l'effort d'assigner manuellement des nombres uniques lors de l'encodage.		
comments	Tableau associatif de commentaires de fichier. Peut être traduit par <code>strtoupper(\$name) . "=\$value"</code> . Note : Cette option de contexte n'est pas disponible en oggvorbis-0.1	Écriture uniquement	<code>array('ENCODER' => 'PHP/OggVorbis, http://pear.php.net/oggvorbis')</code>

8.89.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Constante	Définition
OGGVORBIS_PCM_U8	PCM 8-bit non-signé.
OGGVORBIS_PCM_S8	PCM 8-bit signé.
OGGVORBIS_PCM_U16_LE	PCM 16-bit non-signé. Arrangements normaux Little Endian .
OGGVORBIS_PCM_U16_BE	PCM 16-bit non-signé. Arrangements normaux Big Endian .
OGGVORBIS_PCM_S16_LE	PCM 16-bit signé. Arrangements normaux Little Endian .
OGGVORBIS_PCM_S16_BE	PCM 16-bit signé. Arrangements normaux Big Endian .

- [Introduction](#)
- [Pré-requis](#)
- [Configuration à l'exécution](#)
- [Installation](#)
- [Options de contexte](#)
- [Constantes pré-définies](#)

8.90 Gestion Audio OpenAL

8.90.1 Introduction

Plate-forme indépendante pour la gestion de l'audio. Requièrre la bilibliothèque OpenAL .

8.90.2 Installation

Cette extension PECL n'est pas intégrée à PHP.

Des informations comme les notes sur les nouvelles versions, les téléchargements, les sources des fichiers, les informations concernant les mainteneurs ainsi qu'un CHANGELOG, peuvent être trouvées ici : <http://pecl.php.net/package/openal> .

Vous pouvez télécharger ces bibliothèques DLL pour ces extensions PECL depuis la page de téléchargement de PHP sous sur la page <http://snaps.php.net/> .

8.90.3 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.90.4 Types de ressources

Cette extension définit quatre types de ressource : **Open AL(Device)** - Retournée par openal_device_open , **Open AL(Context)** - Retournée par openal_context_create , **Open AL(Buffer)** - Retournée par openal_buffer_create , and **Open AL(Source)** - Retournée par openal_source_create .

8.90.5 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

ALC_FREQUENCY (entier)
Attribut de contexte
ALC_REFRESH (entier)
Attribut de contexte
ALC_SYNC (entier)
Attribut de contexte
AL_FREQUENCY (entier)
Configuration du buffer
AL_BITS (entier)
Configuration du buffer
AL_CHANNELS (entier)
Configuration du buffer
AL_SIZE (entier)
Configuration du buffer
AL_BUFFER (entier)

Configuration de la source/de l'écoute (Entier)
AL_SOURCE_RELATIVE (entier)
Configuration de la source/de l'écoute (Entier)
AL_SOURCE_STATE (entier)
Configuration de la source/de l'écoute (Entier)
AL_PITCH (entier)
Configuration de la source/de l'écoute (Nombre à virgule flottante)
AL_GAIN (entier)
Configuration de la source/de l'écoute (Nombre à virgule flottante)
AL_MIN_GAIN (entier)
Configuration de la source/de l'écoute (Nombre à virgule flottante)
AL_MAX_GAIN (entier)
Configuration de la source/de l'écoute (Nombre à virgule flottante)
AL_MAX_DISTANCE (entier)
Configuration de la source/de l'écoute (Nombre à virgule flottante)
AL_ROLLOFF_FACTOR (entier)
Configuration de la source/de l'écoute (Nombre à virgule flottante)
AL_CONE_OUTER_GAIN (entier)
Configuration de la source/de l'écoute (Nombre à virgule flottante)
AL_CONE_INNER_ANGLE (entier)
Configuration de la source/de l'écoute (Nombre à virgule flottante)
AL_CONE_OUTER_ANGLE (entier)
Configuration de la source/de l'écoute (Nombre à virgule flottante)
AL_REFERENCE_DISTANCE (entier)
Configuration de la source/de l'écoute (Nombre à virgule flottante)
AL_POSITION (entier)
Configuration de la source/de l'écoute (Nombre vectoriel à virgule flottante)
AL_VELOCITY (entier)
Configuration de la source/de l'écoute (Nombre vectoriel à virgule flottante)
AL_DIRECTION (entier)
Configuration de la source/de l'écoute (Nombre vectoriel à virgule flottante)
AL_ORIENTATION (entier)
Configuration de la source/de l'écoute (Nombre vectoriel à virgule flottante)
AL_FORMAT_MONO8 (entier)
Format PCM
AL_FORMAT_MONO16 (entier)
Format PCM
AL_FORMAT_STEREO8 (entier)
Format PCM
AL_FORMAT_STEREO16 (entier)
Format PCM
AL_INITIAL (entier)
État de la Source
AL_PLAYING (entier)
État de la Source
AL_PAUSED (entier)
État de la Source
AL_STOPPED (entier)
État de la Source
AL_LOOPING (entier)
État de la Source
AL_TRUE (entier)
Valeur booléen reconnue par OpenAL
AL_FALSE (entier)

Valeur booléen reconnue par OpenAL

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [openal_buffer_create](#)
- [openal_buffer_data](#)
- [openal_buffer_destroy](#)
- [openal_buffer_get](#)
- [openal_buffer_loadwav](#)
- [openal_context_create](#)
- [openal_context_current](#)
- [openal_context_destroy](#)
- [openal_context_process](#)
- [openal_context_suspend](#)
- [openal_device_close](#)
- [openal_device_open](#)
- [openal_listener_get](#)
- [openal_listener_set](#)
- [openal_source_create](#)
- [openal_source_destroy](#)
- [openal_source_get](#)
- [openal_source_pause](#)
- [openal_source_play](#)
- [openal_source_rewind](#)
- [openal_source_set](#)
- [openal_source_stop](#)
- [openal_stream](#)

8.90.7 openal_buffer_data() : Charge un buffer avec des données

bool **openal_buffer_data** (resource buffer , int format , string data , int freq)

8.90.8 openal_buffer_destroy() : Détruit un buffer OpenAL

bool **openal_buffer_destroy** (resource buffer)

8.90.9 openal_buffer_get() : Récupère les propriétés du buffer OpenAL

int **openal_buffer_get** (resource buffer , int property)

8.90.10 openal_buffer_loadwav() : Charge un fichier .wav dans le buffer

bool **openal_buffer_loadwav** (resource buffer , string wavfile)

8.90.11 `openal_context_create()` : Crée un contexte de traitement audio

resource `openal_context_create` (resource device)

8.90.12 `openal_context_current()` : Rend courant le contexte spécifié

bool `openal_context_current` (resource context)

8.90.13 `openal_context_destroy()` : Détruit un contexte

bool `openal_context_destroy` (resource context)

8.90.14 `openal_context_process()` : Traite le contexte spécifié

bool `openal_context_process` (resource context)

8.90.15 `openal_context_suspend()` : Suspend le contexte spécifié

bool `openal_context_suspend` (resource context)

8.90.16 `openal_device_close()` : Ferme un périphérique OpenAL

bool `openal_device_close` (resource device)

8.90.17 `openal_device_open()` : Initialise une interface audio OpenAL

resource `openal_device_open` (*string* device_desc)

8.90.18 `openal_listener_get()` : Récupère une propriété d'auditeur

mixed `openal_listener_get` (int property)

8.90.19 `openal_listener_set()` : Définit une propriété d'auditeur

bool `openal_listener_set` (int property , mixed setting)

8.90.20 `openal_source_create()` : Génère une ressource de source

resource `openal_source_create` (void)

8.90.21 `openal_source_destroy()` : Détruit une ressource de source

`bool openal_source_destroy (resource source)`

8.90.22 `openal_source_get()` : Récupère une propriété de source OpenAL

`mixed openal_source_get (resource source , int property)`

8.90.23 `openal_source_pause()` : Marque une pause dans la source

`bool openal_source_pause (resource source)`

8.90.24 `openal_source_play()` : Démarre la lecture de la source

`bool openal_source_play (resource source)`

8.90.25 `openal_source_rewind()` : Revient en arrière dans la source

`bool openal_source_rewind (resource source)`

8.90.26 `openal_source_set()` : Définit une propriété de la source

`bool openal_source_set (resource source , int property , mixed setting)`

8.90.27 `openal_source_stop()` : Arrête la lecture de la source

`bool openal_source_stop (resource source)`

8.90.28 `openal_stream()` : Démarre le streaming d'une source

`resource openal_stream (resource source , int format , int rate)`

8.91 OpenSSL

8.91.1 Introduction

Cette extension utilise les fonctions de [OpenSSL](#) pour générer et vérifier les signatures, ainsi que pour sceller (chiffrer) et ouvrir (déchiffrer) les données. OpenSSL offre beaucoup de fonctionnalités que ce module n'offre pas actuellement. Quelques-unes pourront être ajoutées dans le futur.

8.91.2 Pré-requis

Afin de pouvoir utiliser les fonctions OpenSSL, vous devez installer les fonctions [OpenSSL](#). PHP depuis les version 4.0.5 et 4.3.1 fonctionnent avec openssl >= 0.9.5. Les autres versions (PHP <=4.0.4p11 et >= 4.3.2) nécessitent OpenSSL >= 0.9.6.

Attention

Vous êtes vivement encouragé à utiliser la version la plus récente d'OpenSSL afin d'éviter certaines vulnérabilités sur votre serveur web.

8.91.3 Installation

Pour utiliser le support OpenSSL de PHP, vous devez aussi compiler PHP avec l'option de configuration `--with-openssl[=DIR]`.

Note

Note aux utilisateurs Win32

Afin d'activer ce module dans l'environnement Windows, vous devez copier les bibliothèques **libeay32.dll** depuis le dossier DLL de PHP/Win32 dans le dossier système SYSTEM32 de votre machine (par exemple : C:\WINNT\SYSTEM32 ou C:\WINDOWS\SYSTEM32).

De plus, si vous envisagez de générer des clés et de signer des messages, vous devez installer un fichier openssl.cnf valide sur votre système. Depuis PHP 4.3.0, une configuration simple est incluse dans le dossier openssl de la distribution Windows. Si vous utilisez PHP PHP 4.2.0 ou plus ancien, et que ces fichiers manquent, vous pouvez les télécharger sur [le site d'OpenSSL](#) ou en téléchargeant les fichiers de configuration de PHP 4.3.0.

PHP va rechercher le fichier openssl.cnf suivant la tactique suivante :

- La variable d'environnement **OPENSSL_CONF**, si elle est définie, sera utilisée comme chemin (comprenant le fichier) vers le fichier de configuration.
- La variable d'environnement **SSLEAY_CONF**, si elle est définie, sera utilisée comme chemin (comprenant le fichier) vers le fichier de configuration.
- Le fichier openssl.cnf sera supposé se trouver dans le dossier des certificats, tel que configuré lors de la compilation de la bibliothèque openssl. Cela signifie généralement c:\usr\local\ssl\openssl.cnf.

Dans votre installation, vous devrez décider si vous allez installer le fichier dans c:\usr\local\ssl\openssl.cnf ou si vous allez le faire ailleurs et configurer une variable d'environnement (possiblement par site virtuel). Notez qu'il est possible de remplacer le chemin par défaut en utilisant le paramètre configargs des fonctions qui requièrent un fichier de configuration.

8.91.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.91.5 Types de ressources

8.91.6 Paramètres clés/certificats

Un bon nombre de fonctions OpenSSL demandent une clé et un certificat comme paramètres. PHP 4.0.5 et plus récent utilisait des clés ou certificats sous forme de ressource, retournée par l'une des fonctions `openssl_get_xxx()`. Les versions ultérieures utilisent l'une des méthodes suivantes :

- Certificats
 1. Une ressource X.509 retournée par `openssl_x509_read`
 2. Une chaîne au format `file://path/to/cert.pem` ; Le fichier ainsi repéré doit contenir un certificat, encodé au format PEM
 3. Une chaîne contenant le contenu d'un certificat, encodé au format PEM.
- Clés publiques/privées
 1. Une ressource clé, retournée par la fonction `openssl_get_publickey` ou `openssl_get_privatekey`
 2. Pour les clés publiques seulement : une ressource X.509
 3. Une chaîne avec le format : `file://path/to/file.pem` . Le fichier doit contenir une clé privée, ou un certificat, encodé au format PEM (il peut contenir les deux).
 4. Une chaîne contenant une clé ou un certificat encodé au format PEM
 5. Pour les clés privées, vous pouvez aussi utiliser la syntaxe `array($key, $passphrase)` , où `$key` représente une clé spécifiée par un fichier ou une représentation textuelle comme cité ci-dessus, et `$passphrase` représente une chaîne contenant la passe-phrase de cette clé privée.

8.91.7 Vérification de certificats

Lorsque vous appelez une fonction qui va vérifier une signature ou un certificat, le paramètre `cainfo` doit être un tableau contenant les noms d'un dossier et d'un fichier indiquant les tiers de confiance. Si un dossier est spécifié, il doit être correct, car `openssl` va l'utiliser.

8.91.8 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

8.91.8.1 Options de validations générales

`X509_PURPOSE_SSL_CLIENT` (entier)
`X509_PURPOSE_SSL_SERVER` (entier)
`X509_PURPOSE_NS_SSL_SERVER` (entier)
`X509_PURPOSE_SMIME_SIGN` (entier)
`X509_PURPOSE_SMIME_ENCRYPT` (entier)
`X509_PURPOSE_CRL_SIGN` (entier)

X509_PURPOSE_ANY ([entier](#))**8.91.8.2 Options de remplissage (Padding)**

OPENSSL_PKCS1_PADDING ([entier](#))
 OPENSSL_SSLV23_PADDING ([entier](#))
 OPENSSL_NO_PADDING ([entier](#))
 OPENSSL_PKCS1_OAEP_PADDING ([entier](#))

8.91.8.3 Types de clés

OPENSSL_KEYTYPE_RSA ([entier](#))
 OPENSSL_KEYTYPE_DSA ([entier](#))
 OPENSSL_KEYTYPE_DH ([entier](#))

8.91.8.4 Constantes/options PKCS7

Les fonctions S/MIME utilisent des options qui sont spécifiées par un champ de bits. Les valeurs valides sont :

Constante	Description
PKCS7_TEXT	Ajoute le texte plein en clair dans les en-têtes du message signé/chiffré. Lors du déchiffrement ou la vérification, il supprime purement et simplement ces données. Si le message chiffré ou signé n'est pas du type MIME, une erreur surviendra.
PKCS7_BINARY	Normalement, le message est converti au format canonique qui utilise effectivement des CR et LF comme fin de ligne, comme demandé dans les spécifications de S/MIME. Lorsque cette option est activée, le message ne sera pas converti. Cela sert lorsque vous manipulez des données binaires qui ne sont pas au format MIME.
PKCS7_NOINTERN	Lors de la vérification d'un message, les certificats (s'il y en a) inclus dans le message sont normalement utilisés pour rechercher le certificat de signature. Avec cette option, seul le certificat spécifié par le paramètre <code>extracerts</code> de la fonction <code>openssl_pkcs7_verify</code> est utilisé. Les certificats fournis peuvent toujours être utilisés, avec un niveau de confiance réduit.
PKCS7_NOVERIFY	Ne vérifie pas les certificats des signataires d'un message signé.
PKCS7_NOCHAIN	N'enchaîne pas les vérifications des signataires de certificats. C'est-à-dire, n'utilise pas les certificats contenu dans le message.
PKCS7_NOCERTS	Lors de la signature d'un message, le certificat du signataire est normalement inclus. Avec cette option, c'est désactivé. Cela va réduire la taille du message, mais le vérificateur devra avoir une copie locale du certificat du signataire (passée au paramètre <code>extracerts</code> , avec la fonction <code>openssl_pkcs7_verify</code>).
PKCS7_NOATTR	Normalement, lorsqu'un message est signé, un jeu d'attributs contenant l'heure de signature et l'algorithme symétrique supporté, est inclus dans le message. Avec cette option, il n'est pas inclus.
PKCS7_DETACHED	Lors de la signature d'un message, utilise la signature en texte clair, avec le type MIME "multipart/signed". C'est la valeur par défaut du paramètre <code>flags</code> pour la fonction <code>openssl_pkcs7_sign</code> . Si vous annulez cette option, le message sera signé de manière opaque, ce qui résiste mieux à la traduction

	des relais emails (certains serveurs mail anciens corrompent les messages), mais empêche la lecture par les client emails qui ne connaissent pas S/MIME.
PKCS7_NOSIGS	Ne vérifie pas les signatures d'une message
Note	
Ces constantes ont été ajoutées en PHP 4.0.6.	

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Paramètres clés/certificats](#)
- [Vérification de certificats](#)
- [Constantes pré-définies](#)
- [openssl_csr_export_to_file](#)
- [openssl_csr_export](#)
- [openssl_csr_new](#)
- [openssl_csr_sign](#)
- [openssl_error_string](#)
- [openssl_free_key](#)
- [openssl_get_privatekey](#)
- [openssl_get_publickey](#)
- [openssl_open](#)
- [openssl_pkcs7_decrypt](#)
- [openssl_pkcs7_encrypt](#)
- [openssl_pkcs7_sign](#)
- [openssl_pkcs7_verify](#)
- [openssl_pkey_export_to_file](#)
- [openssl_pkey_export](#)
- [openssl_pkey_free](#)
- [openssl_pkey_get_private](#)
- [openssl_pkey_get_public](#)
- [openssl_pkey_new](#)
- [openssl_private_decrypt](#)
- [openssl_private_encrypt](#)
- [openssl_public_decrypt](#)
- [openssl_public_encrypt](#)
- [openssl_seal](#)
- [openssl_sign](#)
- [openssl_verify](#)
- [openssl_x509_check_private_key](#)
- [openssl_x509_checkpurpose](#)
- [openssl_x509_export_to_file](#)
- [openssl_x509_export](#)
- [openssl_x509_free](#)
- [openssl_x509_parse](#)
- [openssl_x509_read](#)

8.91.10 openssl_csr_export() : Exporte un CSR vers un fichier ou une variable

bool **openssl_csr_export** (resource csr , string out , bool notext)

openssl_csr_export convertit la requête de signature de certificat représentée par csr au format ASCII et la stocke dans la variable out , qui est passée par référence.

Le paramètre optionnel notext affecte le niveau de détails du résultat. S'il vaut FALSE , alors des informations supplémentaires lisibles seront incluses dans le résultat. La valeur par défaut de notext est TRUE .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi openssl_csr_export_to_file , openssl_csr_new et openssl_csr_sign .

8.91.11 openssl_csr_new() : Génère une CSR

mixed **openssl_csr_new** (array dn , resource privkey , array configargs , array extraattribs)

openssl_csr_new génère une nouvelle CSR (Certificate Signing Request , requête de signature de certificat), basée sur les informations fournies par dn , qui représente le nom unique (Distinguished Name) qui doit être utilisé pour ce certificat.

privkey doit être une clé privée qui a été générée par openssl_pkey_new (ou obtenue autrement par l'une des fonctions de la famille openssl_pkey) La clé publique sera utilisée pour signer la CSR.

extraattribs est utilisé pour spécifier des options de configuration supplémentaires pour la CSR. Les deux paramètres dn et extraattribs sont des tableaux associatifs dont les clés sont converties en OIDs et appliquées aux parties appropriées de la requête.

Note
Vous devez installer un fichier openssl.cnf valide pour que cette fonction opère correctement. Voyez les notes dans la <u>section installation</u> pour plus d'informations.

Par défaut, les informations du fichier openssl.conf de votre système sont utilisées pour initialiser la requête ; vous pouvez spécifier une section du fichier de configuration en utilisant l'index config_section_section du tableau configargs . Vous pouvez aussi spécifier une configuration OpenSSL alternative en utilisant l'index config , et en lui donnant la valeur du chemin à utiliser. Les index suivants, si fournis dans le paramètre configargs , se comportent comme leur équivalent du fichier openssl.conf , tel qu'indiqué ci-dessous.

Index configargs	Type	Equivalent openssl.conf	Description
digest_alg	<u>chaîne de caractères</u>	default_md	Sélectionne la méthode "digest" à utiliser
x509_extensions	<u>chaîne de caractères</u>	x509_extensions	Sélectionne quelles extensions doivent être utilisées lors de la création du certificat x509
req_extensions	<u>chaîne de caractères</u>	req_extensions	Sélectionne quelles extensions doivent être utilisées lors de la création d'une CSR
private_key_bits	<u>entier</u>	default_bits	Spécifie le nombre de bits à utiliser lors de la génération de la clé privée
private_key_type	<u>entier</u>	none	Spécifie le type de clé privée à générer. Cela peut être l'une des constantes suivantes : OPENSSL_KEYTYPE_DSA ,

			OPENSSL_KEYTYPE_DH ou OPENSSL_KEYTYPE_RSA . La valeur par défaut est OPENSSL_KEYTYPE_RSA , ce qui est le seul type de clé actuellement supporté.
encrypt_key	booléen	encrypt_key	Doit être une clé exportée (avec mot de passe) qui sera chiffrée?

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec `openssl_csr_new` - création d'un certificat auto chiffré

```
<?php
// Assigne les valeurs du nom distingué à utiliser avec le certificat
// Vous devez remplacer les valeurs suivantes pour qu'elles correspondent
// au nom de votre compagnie, ou, plus précisément, le nom de la personne
// qui représente le site de votre compagnie pour qui vous générez des
// clés. Pour les certificats SSL, le commonName est généralement le nom de domaine
// pour lequel vous installez le certificat, mais pour les certificats
// le commonName sera le nom de la personne qui utilisera le certificat.
$dn = array(
    "countryName" => "UK",
    "stateOrProvinceName" => "Somerset",
    "localityName" => "Glastonbury",
    "organizationName" => "The Brain Room Limited",
    "organizationalUnitName" => "PHP Documentation Team",
    "commonName" => "Wez Furlong",
    "emailAddress" => "wez@php.net"
);

// Génère les clés privée et publique
$privkey = openssl_pkey_new();

// Génère la requête de signature de certificat
$csr = openssl_csr_new($dn, $privkey);

// Vous souhaitez généralement créer un certificat auto chiffré
// une fois que votre autorité de certification accède à votre requête
// Cette commande crée un certificat auto chiffré
$$sscert = openssl_csr_sign($csr, null, $privkey, 365);

// Maintenant, vous voulez préserver la clé secrète, la CSR et le certificat
// auto chiffré, de façon à ce qu'ils puissent être installés sur votre
// serveur web, serveur mail ou client mail (suivant l'utilisation).
// Cet exemple vous montre comment placer ces éléments dans des variables
// mais vous pouvez aussi les mettre directement dans des fichiers.
// Typiquement, vous allez envoyer la CSR à votre autorité de certification
// qui vous émettra un "vrai" certificat.
openssl_csr_export($csr, $csrout) and var_dump($csrout);
openssl_x509_export($sscert, $certout) and var_dump($certout);
openssl_pkey_export($privkey, $pkeyout, "mypassword") and var_dump($pkeyout);

// Affiche les erreurs qui sont survenues
while (($e = openssl_error_string()) !== false) {
    echo $e . "\n";
}
?>
```

8.91.12 `openssl_csr_sign()` : Signe un CSR avec un autre certificat

resource `openssl_csr_sign` (mixed csr , mixed cacert , mixed priv_key , int days , array configargs , int serial)

`openssl_csr_sign` génère un certificat x509 (une ressource) depuis la CSR `csr`, générée précédemment par `openssl_csr_new`, mais ce paramètre peut aussi être une CSR encodée au format PEM, et spécifiée avec `file://path/to/csr` ou une chaîne exportée par `openssl_csr_export`. Le certificat généré sera signé par le certificat `cacert`. Si `cacert` vaut `NULL`, le certificat généré sera auto-signé. `priv_key` est la clé privée qui correspond au certificat `cacert`. `days` spécifie la durée pour laquelle le certificat est valide, en nombre de jours. Vous pouvez affiner la signature CSR avec `configargs`. Voir la fonction `openssl_csr_new` pour plus d'informations sur le paramètre `configargs`. Depuis PHP 4.3.3, vous pouvez spécifier le numéro de série du certificat généré avec l'argument `serial`. Dans les versions précédentes, il vaut toujours 0.

`openssl_csr_sign` retourne une ressource de certificat x509 en cas de succès, et `FALSE` sinon.

Note

Vous devez installer un fichier `openssl.cnf` valide pour que cette fonction opère correctement. Voyez les notes dans la [section installation](#) pour plus d'informations.

Exemple avec `openssl_csr_sign` : signer une CSR (comment réaliser votre propre autorité de certification)

```
<?php
// Supposons que ce script est configuré pour recevoir des CSR qui ont
// été collés dans un champ textarea depuis une autre page
$csrdata = $_POST["CSR"];

// Nous allons signer la requête avec notre propre certificat, en tant
// qu'"autorité de certification". Vous pouvez utiliser n'importe quel
// certificat pour en signer un autre, mais le processus est inutile à moins
// que le certificat de signature n'ait la confiance des utilisateurs
// qui utiliseront le nouveau certificat signé.

// Nous avons besoin de notre certificat et de la clé privée
$cacert = "file://path/to/ca.crt";
$privkey = array("file://path/to/ca.key", "la_cle_secrete_de_votre_certificat");

$usercert = openssl_csr_sign($csrdata, $cacert, $privkey, 365);

// Affichons maintenant le certificat généré, de façon à ce que l'utilisateur
// puisse le copier/coller dans sa configuration locale (comme un
// fichier qui contient les certificats de son serveur SSL)
openssl_x509_export($usercert, $certout);
echo $certout;

// Affiche toutes les erreurs survenues
while (($e = openssl_error_string()) !== false) {
    echo $e . "\n";
}
?>
```

8.91.13 `openssl_error_string()` : Retourne le message d'erreur OpenSSL

string `openssl_error_string` (void)

`openssl_error_string` retourne un message d'erreur, sous forme de chaîne de caractères, ou `FALSE` s'il n'y a pas de message d'erreur.

`openssl_error_string` retourne la dernière erreur de la bibliothèque OpenSSL. Les messages d'erreurs sont empilés, et `openssl_error_string` doit être appelé plusieurs fois pour afficher toutes les erreurs.

Exemple avec `openssl_error_string`

```
<?php
// Imaginons que vous avez appelé une fonction qui a émis une erreur
while ($msg = openssl_error_string())
    echo $msg . "<br />\n";
?>
```

8.91.14 `openssl_free_key()` : Libère les ressources

void **openssl_free_key** (resource key_identifier)

`openssl_free_key` libère les ressources associées à `key_identifier` .

8.91.15 `openssl_get_privatekey()` : Alias de `openssl_pkey_get_private`

Cette fonction est un alias de : `openssl_pkey_get_private` .

8.91.16 `openssl_get_publickey()` : Alias de `openssl_pkey_get_public`

Cette fonction est un alias de : `openssl_pkey_get_public` .

8.91.17 `openssl_open()` : Ouvre des données scellées

bool **openssl_open** (string sealed_data , string open_data , string env_key , mixed priv_key_id)

`openssl_open` ouvre (déchiffre) les données `sealed_data` en utilisant la clé privée `priv_key_id` et la clé d'enveloppe `env_key` et remplit `open_data` avec les données déchiffrées. La clé d'enveloppe est générée lorsque les données sont scellées, et ne peut être utilisée qu'avec la clé privée spécifique. Reportez-vous à `openssl_seal` pour plus d'informations.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. En cas de succès, les données déchiffrées sont placées dans `open_data` .

Exemple avec `openssl_open`

```
<?php
// On suppose que $sealed et $env_key contiennent les données scellées
// et la clé d'enveloppe, fournies par l'expéditeur
// lecture de la clé privée dans un fichier
$fp = fopen("/src/openssl-0.9.6/demos/sign/key.pem", "r");
$priv_key = fread($fp, 8192);
fclose($fp);
$pkeyid = openssl_get_privatekey($priv_key);

// déchiffrement des données : elles sont placées dans $open
if (openssl_open($sealed, $open, $env_key, $pkeyid)) {
    echo "Voici les données déchiffrées : ", $open;
} else {
    echo "Impossible de déchiffrer les données";
}

// libération des ressources
openssl_free_key($pkeyid);
```

```
?>
```

Voir aussi [openssl_seal](#) .

8.91.18 openssl_pkcs7_decrypt() : Déchiffre un message S/MIME

bool **openssl_pkcs7_decrypt** (string infilename , string outfilename , mixed recipcert , *mixed recipkey*)

`openssl_pkcs7_decrypt` déchiffre le message S/MIME contenu dans le fichier infilename , en utilisant le certificat et la clé privée spécifiés par recipcert et recipkey . Le message déchiffré sera écrit dans le fichier outfilename .

Exemple avec `openssl_pkcs7_decrypt`

```
<?php
// $cert et $key contiennent vos certificats et clés privés
// On suppose aussi que le message vous est destiné
$infilename = "message_chiffre.msg";

// Le message chiffré
$outfilename = "message_dechiffre.msg";

// Assurez-vous de bien pouvoir écrire dans ce fichier
if (openssl_pkcs7_decrypt($infilename, $outfilename, $cert, $key)) {
    echo 'déchiffré!';
} else {
    echo 'impossible de déchiffrer!';
}
?>
```

8.91.19 openssl_pkcs7_encrypt() : Chiffre un message S/MIME

bool **openssl_pkcs7_encrypt** (string infile , string outfile , mixed recipcerts , array headers , *int flags* , *int cipherid*)

`openssl_pkcs7_encrypt` prend le contenu du fichier infilename et le chiffre en utilisant un chiffrement RC2 à 40-bit, de manière à ce que le message ne puisse être lu que par le possesseur de recipcerts , qui peut être un certificat X.509 ou un tableau de certificats X.509. headers est un tableau d'en-têtes qui seront ajoutés en tête de message, une fois que les données auront été chiffrées. flags peut être utilisé pour spécifier des options qui affecteront le chiffrement (voir les [constantes PKCS7](#)). headers peut être un tableau associatif, dont les clés sont les noms d'en-tête, ou bien un tableau indexé dont chaque ligne contient un en-tête complet. Cipher peut être sélectionné avec le paramètre cipherid depuis PHP 5.

Exemple avec `openssl_pkcs7_encrypt`

```
<?php
// le message que vous souhaitez chiffrer et envoyer à votre agent secret
// en mission commandée, appelé "nighthawk". Vous avez son certificat
// dans le fichier "nighthawk.pem"
$data = <<<EOD
Nighthawk,

Top secret, uniquement votre lecture !

L'ennemi approche! Rendez-vous au café à 8 h 30,
pour votre faux passeport.
```

```

HQ
EOD;

// Chargement de la clé
$key = file_get_contents("nighthawk.pem");

// sauvez le message dans un fichier
$fp = fopen("msg.txt", "w");
fwrite($fp, $data);
fclose($fp);

// chiffrez le
if (openssl_pkcs7_encrypt("msg.txt", "enc.txt", $key,
    array("To" => "nighthawk@agent.com", // keyed syntax
          "From: HQ <hq@cia.com>", // indexed syntax
          "Subject" => "Secret défense"))) {
    // message chiffré : envoyez-le!
    exec(ini_get("sendmail_path") . " < enc.txt");
}
?>

```

8.91.20 openssl_pkcs7_sign() : Signe un message S/MIME

bool **openssl_pkcs7_sign** (string infilename , string outfilename , mixed signcert , mixed privkey , array headers , int flags , string extracerts)

openssl_pkcs7_sign prend le contenu du fichier infilename et le signe en utilisant le certificat et la clé privée contenus dans les arguments signcert et privkey .

headers est un tableau d'en-têtes qui seront ajoutés aux données chiffrées (voir la fonction openssl_pkcs7_encrypt pour plus de détails sur le format du paramètre).

flags sert à modifier le message final. Voyez les constantes PKCS7 . Par défaut, la valeur est : PKCS7_DETACHED.

extracerts spécifie le nom du fichier contenant un ensemble de certificats supplémentaires à inclure dans la signature, qui pourront aider le destinataire à vérifier les données que vous utilisez.

Exemple avec openssl_pkcs7_sign

```

<?php
// le message que vous voulez signer, afin que le destinataire soit sûr qu'il
// vient bien de vous
$data = <<<EOD

Tu peux dépenser jusqu'à 10000 euros en note de frais.

Ton boss
HQ
EOD;
// sauvez le message dans un fichier
$fp = fopen("msg.txt", "w");
fwrite($fp, $data);
fclose($fp);
// chiffrez le
if (openssl_pkcs7_sign("msg.txt", "signed.txt", "mycert.pem",
    array("file://mycert.pem", "mypassphrase"),
    array("To" => "joes@sales.com", // syntaxe à clé
          "From: HQ <ceo@sales.com>", // syntaxe indexée
          "Subject" => "Secret défense")
    )) {

```

```
// message signed - send it!
exec(ini_get("sendmail_path") . " < signed.txt");
}
?>
```

8.91.21 openssl_pkcs7_verify() : Vérifie la signature d'un message S/MIME

mixed **openssl_pkcs7_verify** (string filename , int flags , string outfilename , array cainfo , string extracerts)

openssl_pkcs7_verify lit le message S/MIME contenu dans le fichier filename et examine la signature digitale. openssl_pkcs7_verify retourne TRUE si la signature est vérifiée, et FALSE sinon (le message a été modifié, ou bien le certificat de signature est invalide). openssl_pkcs7_verify retourne -1 en cas d'erreur de vérification (la vérification s'est mal déroulée, aucune conclusion possible).

flags sert à modifier le message final. Voyez les constantes PKCS7 . Par défaut, la valeur est : PKCS7_DETACHED.

Si le paramètre outfilename est spécifié, il doit être une chaîne contenant le nom d'un fichier qui contient le certificat du signataire, au format PEM.

Si le paramètre cainfo est spécifié, il doit contenir les informations sur les tiers de certificats de confiance utilisé lors de la vérification. Voyez vérification des certificats pour plus de détails.

Si le paramètre extracerts est spécifié, il doit représenter le nom d'un fichier contenant un ensemble de certificats utilisés comme certificats de peu de confiance.

8.91.22 openssl_pkey_export_to_file() : Sauve une clé au format ASCII dans un fichier

bool **openssl_pkey_export_to_file** (mixed key , string outfilename , string passphrase , array configargs)

openssl_pkey_export_to_file sauve la clé au format ASCII (PEM) key dans le fichier outfilename . La clé peut être optionnellement protégée par le mot de passe passphrase . configargs peut être utilisé pour calibrer le processus d'exportation en spécifiant ou remplaçant les options du fichier de configuration d'OpenSSL. Voyez openssl_csr_new pour plus d'informations sur configargs . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Vous devez installer un fichier openssl.cnf valide pour que cette fonction opère correctement. Voyez les notes dans la section installation pour plus d'informations.

8.91.23 openssl_pkey_export() : Lit une représentation exportable de la clé dans une chaîne ou un fichier

bool **openssl_pkey_export** (mixed key , string out , string passphrase , array configargs)

`openssl_pkey_export` exporte la clé `key` sous forme de chaîne au format PEM, et la stocke dans la variable `out` (qui est passée par référence). La clé est éventuellement protégée par le mot de passe `passphrase`. `configargs` peut être utilisé pour calibrer le processus d'exportation en spécifiant ou remplaçant les options du fichier de configuration d'OpenSSL. Voyez `openssl_csr_new` pour plus d'informations sur `configargs`. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Note

Vous devez installer un fichier `openssl.cnf` valide pour que cette fonction opère correctement. Voyez les notes dans la [section installation](#) pour plus d'informations.

8.91.24 `openssl_pkey_free()` : Libère une clé privée

`void openssl_pkey_free (resource key)`

Cette fonction libère une clé privée créée par la fonction `openssl_pkey_new`.

8.91.25 `openssl_pkey_get_private()` : Lit une clé privée

`resource openssl_pkey_get_private (mixed key , string passphrase)`

`openssl_pkey_get_private` retourne une ressource de clé privée en cas de succès, et `FALSE` sinon.

`openssl_pkey_get_private` analyse la clé `key` et la prépare pour être utilisée par d'autres fonctions. `key` peut être l'une des valeurs suivantes :

1. une chaîne au format `file://path/to/file.pem`. Le fichier ainsi désigné doit contenir une clé privée ou un certificat au format PEM (éventuellement les deux).
2. Une clé privée au format PEM.

Le paramètre optionnel `passphrase` doit être utilisé si la clé spécifiée est chiffrée (protégée par un mot de passe).

8.91.26 `openssl_pkey_get_public()` : Extrait une clé privée d'un certificat, et la prépare

`resource openssl_pkey_get_public (mixed certificate)`

`openssl_pkey_get_public` retourne une ressource de clé en cas de succès, et `FALSE` sinon.

`openssl_get_publickey` extrait la clé publique du certificat `certificate` et la prépare pour être utilisée par les autres fonctions. `certificate` peut avoir l'une des valeurs suivantes :

1. une ressource d'un certificat X.509.
2. une chaîne au format `file://path/to/file.pem`. Le fichier ainsi désigné doit contenir une clé privée ou un certificat au format PEM (éventuellement les deux).
3. Une clé privée au format PEM.

8.91.27 openssl_pkey_new() : Génère une nouvelle clé privée

resource **openssl_pkey_new** (*array configargs*)

openssl_pkey_new génère une nouvelle paire de clés. La clé privée est accessible via la fonction openssl_pkey_get_public . Vous pouvez calibrer la génération de la clé (comme le nombre de bits) en utilisant le paramètre *configargs* . Voir la fonction openssl_csr_new pour plus de détails sur *configargs* .

Note

Vous devez installer un fichier `openssl.cnf` valide pour que cette fonction opère correctement. Voyez les notes dans la [section installation](#) pour plus d'informations.

8.91.28 openssl_private_decrypt() : Déchiffre des données avec une clé privée

bool **openssl_private_decrypt** (string *data* , string *decrypted* , mixed *key* , int *padding*)

openssl_private_decrypt déchiffre *data* qui a été chiffrée précédemment avec openssl_private_encrypt , puis stocke le résultat dans la variable *decrypted* . *key* doit être la clé privée utilisée pour chiffrer les données. *padding* vaut par défaut `OPENSSL_PKCS1_PADDING` , mais peut aussi prendre les valeurs de `OPENSSL_SSLV23_PADDING` , `OPENSSL_PKCS1_OAEP_PADDING` ou `OPENSSL_NO_PADDING` .

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Vous pouvez utiliser cette fonction pour déchiffrer les données qui sont supposées vous être personnellement adressées.

Voir aussi openssl_public_encrypt et openssl_public_decrypt .

8.91.29 openssl_private_encrypt() : Chiffre des données avec une clé privée

bool **openssl_private_encrypt** (string *data* , string *crypted* , mixed *key* , int *padding*)

openssl_private_encrypt chiffre les données *data* avec la clé privée *key* et stocke le résultat dans *crypted* . Les données chiffrées peuvent être déchiffrées avec la fonction openssl_public_decrypt . Le paramètre *padding* est fixé par défaut à `OPENSSL_PKCS1_PADDING` mais peut prendre la valeur `OPENSSL_NO_PADDING` .

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Cette fonction peut être utilisée pour signer les données (ou leurs chiffrements) pour prouver qu'elles n'ont pas été écrites par une autre personne.

Voir aussi openssl_public_decrypt et openssl_public_encrypt .

8.91.30 `openssl_public_decrypt()` : Déchiffre des données avec une clé publique

`bool openssl_public_decrypt (string data , string decrypted , mixed key , int padding)`

`openssl_public_decrypt` déchiffre les données `data` qui ont été chiffrées avec la fonction `openssl_private_encrypt` et stocke le résultat dans `decrypted` . `key` doit être la clé publique qui a été utilisée pour chiffrer les données. `padding` vaut par défaut `OPENSSL_PKCS1_PADDING` mais peut également valoir `OPENSSL_NO_PADDING` .

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Vous pouvez utiliser cette fonction pour vérifier si le message a été écrit par le propriétaire de la clé privée.

Voir aussi `openssl_private_encrypt` et `openssl_private_decrypt` .

8.91.31 `openssl_public_encrypt()` : Chiffre des données avec une clé publique

`bool openssl_public_encrypt (string data , string crypted , mixed key , int padding)`

`openssl_public_encrypt` chiffre les données `data` avec la clé publique `key` et stocke le résultat dans `crypted` . Les données chiffrées peuvent être déchiffrées avec la fonction `openssl_private_decrypt` . `padding` vaut par défaut `OPENSSL_PKCS1_PADDING` , mais peut valoir `OPENSSL_SSLV23_PADDING` , `OPENSSL_PKCS1_OAEP_PADDING` ou `OPENSSL_NO_PADDING` .

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Cette fonction peut être utilisée pour chiffrer un message qui pourra être lu uniquement par le propriétaire de la clé privée. Elle peut être également utilisée pour stocker des données sécurisées dans une base de données.

Voir aussi `openssl_private_decrypt` et `openssl_private_encrypt` .

8.91.32 `openssl_seal()` : Scelle des données

`int openssl_seal (string data , string sealed_data , array env_keys , array pub_key_ids)`

`openssl_seal` retourne la longueur des données scellées en cas de succès, et `FALSE` sinon. En cas de succès, les données scellées sont placées dans le paramètre `sealed_data` , et les clés d'enveloppe dans `env_keys` .

`openssl_seal` scelle (chiffre) les données `data` en utilisant l'algorithme RC4 avec une clé secrète générée aléatoirement. La clé est chiffrée avec chaque clé publique associée à `pub_key_ids` et chaque clé ainsi chiffrée est retournée dans `env_keys` . Cela signifie que vous pouvez envoyer des données scellées à plusieurs destinataires (en supposant que chacun ait reçu la clé publique). Chaque destinataire doit recevoir les données chiffrées et la clé d'enveloppe, qui a été chiffrée avec la clé publique du destinataire.

Exemple avec `openssl_seal`

```

<?php
// On suppose que $data contient les données à sceller
// lecture de la clé publique pour chaque destinataire
$fp = fopen("/src/openssl-0.9.6/demos/maurice/cert.pem", "r");
$cert = fread($fp, 8192);
fclose($fp);
$pk1 = openssl_get_publickey($cert);
// pour le deuxième destinataire
$fp = fopen("/src/openssl-0.9.6/demos/sign/cert.pem", "r");
$cert = fread($fp, 8192);
fclose($fp);
$pk2 = openssl_get_publickey($cert);

// scelle le message : seuls, les possesseurs de $pk1 et $pk2 peuvent déchiffrer
// le message $sealed avec les clés $keys[0] et $keys[1] (respectivement).
openssl_seal($data, $sealed, $keys, array($pk1,$pk2));

// libère les clés de la mémoire
openssl_free_key($pk1);
openssl_free_key($pk2);
?>

```

Voir aussi [openssl_open](#) .

8.91.33 openssl_sign() : Signe les données

bool **openssl_sign** (string data , string signature , mixed priv_key_id , int signature_alg)

`openssl_sign` calcule la signature des données data en utilisant l'algorithme SHA1 (hashing) suivi du chiffrement avec la clé privée priv_key_id . Notez que les données elles-mêmes ne sont pas chiffrées.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.. En cas de succès, la signature est placée dans signature .

Exemple avec `openssl_sign`

```

<?php
// On suppose que $data contient les données à signer
// lecture de la clé publique pour chaque destinataire
$fp = fopen("/src/openssl-0.9.6/demos/sign/key.pem", "r");
$priv_key = fread($fp, 8192);
fclose($fp);
$pkeyid = openssl_get_privatekey($priv_key);

// calcule de la signature
openssl_sign($data, $signature, $pkeyid);

// libère les clés de la mémoire
openssl_free_key($pkeyid);
?>

```

Voir aussi [openssl_verify](#) .

8.91.34 openssl_verify() : Vérifie une signature

int **openssl_verify** (string data , string signature , mixed pub_key_id)

`openssl_verify` retourne 1 si la signature est correcte, 0 si elle est incorrecte et -1 en cas d'erreur.

`openssl_verify` vérifie que la signature est correcte pour les données `data`, et avec la clé publique `pub_key_id`. Cette clé doit être la clé publique correspondant à la clé privée utilisée lors de la signature.

Exemple avec `openssl_verify`

```
<?php
// On suppose que $data et $signature contiennent les données à signer et
// la signature
// lecture de la clé publique depuis le certificat
$fp = fopen("/src/openssl-0.9.6/demos/sign/cert.pem", "r");
$cert = fread($fp, 8192);
fclose($fp);
$pubkeyid = openssl_get_publickey($cert);

// indique si la signature est correcte
$ok = openssl_verify($data, $signature, $pubkeyid);
if ($ok == 1) {
    echo 'Signature valide';
} elseif ($ok == 0) {
    echo 'Signature erronée';
} else {
    echo 'Erreur de vérification de la signature';
}
// libère les clés de la mémoire
openssl_free_key($pubkeyid);
?>
```

Voir aussi `openssl_sign`.

8.91.35 `openssl_x509_check_private_key()` : Vérifie si une clé privée correspond à un certificat

`bool openssl_x509_check_private_key (mixed cert , mixed key)`

`openssl_x509_check_private_key` retourne TRUE si `key` est la clé privée qui correspond au certificat `cert`, ou FALSE sinon.

8.91.36 `openssl_x509_checkpurpose()` : Vérifie l'usage d'un certificat

`int openssl_x509_checkpurpose (mixed x509cert , int purpose , array cainfo , string untrustedfile)`

`openssl_x509_checkpurpose` TRUE si le certificat peut être utilisé pour un but particulier, FALSE s'il ne le peut pas, et -1 en cas d'erreur.

`openssl_x509_checkpurpose` examine le certificat spécifié par `x509cert`, pour voir s'il peut être utilisé pour une opération particulière `purpose`.

`cainfo` doit être un tableau de dossiers/fichiers de CA de confiance comme décrit dans la [Vérification des certificats](#). Par défaut, c'est un tableau vide.

`untrustedfile`, si spécifié, est le nom d'un fichier au format PEM contenant les certificats qui pourront aider lors de la vérification du certificat, même si une confiance limitée doit leur être portée.

Constante	Description
X509_PURPOSE_SSL_CLIENT	Est-ce que le certificat peut être utilisé avec le client d'une connexion SSL ?
X509_PURPOSE_SSL_SERVER	Est-ce que le certificat peut être utilisé avec le serveur d'une connexion SSL ?
X509_PURPOSE_NS_SSL_SERVER	Est-ce que le certificat peut être utilisé avec un serveur Netscape d'une connexion SSL ?
X509_PURPOSE_SMIME_SIGN	Est-ce que le certificat peut être utilisé pour signer des courriers à la norme S/MIME ?
X509_PURPOSE_SMIME_ENCRYPT	Est-ce que le certificat peut être utilisé pour chiffrer un courrier au format S/MIME ?
X509_PURPOSE_CRL_SIGN	Est-ce que le certificat peut être utilisé pour chiffrer une liste de révocation de certificats ? (CRL)?
X509_PURPOSE_ANY	Est-ce que le certificat peut être utilisé pour n'importe lequel de ces cas ?

Ces options ne sont pas des champs de bits : vous ne pouvez en passer qu'une seule à la fois.

8.91.37 openssl_x509_export_to_file() : Exporte un certificat vers un fichier

`bool openssl_x509_export_to_file (mixed x509 , string outfilename , bool notext)`

`openssl_x509_export_to_file` stocke un certificat x509 dans le fichier outfilename au format PRM.

Le paramètre optionnel notext affecte le niveau de détails du résultat. S'il vaut FALSE , alors des informations supplémentaires lisibles seront incluses dans le résultat. La valeur par défaut de notext est TRUE .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.91.38 openssl_x509_export() : Exporte un certificat vers une variable

`bool openssl_x509_export (mixed x509 , string output , bool notext)`

`openssl_x509_export` stocke un certificat x509 dans le fichier output au format PRM.

Le paramètre optionnel notext affecte le niveau de détails du résultat. S'il vaut FALSE , alors des informations supplémentaires lisibles seront incluses dans le résultat. La valeur par défaut de notext est TRUE .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.91.39 openssl_x509_free() : Libère les ressources prises par un certificat

`void openssl_x509_free (resource x509cert)`

`openssl_x509_free` libère les ressources prises par le certificat x509cert

8.91.40 openssl_x509_parse() : Analyse un certificat X509

array **openssl_x509_parse** (mixed *x509cert* , *bool shortnames*)

openssl_x509_parse analyse le certificat X509 *x509cert* , et retourne les informations contenues dedans, y compris le sujet (*subject*), nom (*name*), émetteur (*issuer name*), dates de début et de fin (*valid from date* et *valid to date*), etc. *shortnames* contrôle l'indexation des données dans le tableau : si *shortnames* vaut TRUE (valeur par défaut), alors les champs seront indexés avec la forme courte des noms, sinon, les noms longs seront utilisés. (par exemple, CN est le nom court de *commonName*).

La structure des données retournées est (délibérément) non documentée, car elle est sujette à des changements probables.

8.91.41 openssl_x509_read() : Analyse un certificat X.509 et retourne une ressource

resource **openssl_x509_read** (mixed *x509certdata*)

openssl_x509_read analyse le certificat *x509certdata* et retourne un identifiant de ressource.

8.92 Oracle [obsolète]

8.92.1 Introduction

Cette extension ajoute le support des bases de données Oracle. Voyez aussi [OCI8](#) .

Note

Cette extension a été déplacée dans le module [PECL](#) et ne sera plus intégrée dans PHP à partir de PHP 5.1.0.

8.92.2 Installation

Vous devez compiler PHP avec l'option `--with-oracle[=DIR]` , où DIR est par défaut la valeur de la variable d'environnement `ORACLE_HOME`.

8.92.3 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

[ORA_BIND_INOUT](#) ([entier](#))
[ORA_BIND_IN](#) ([entier](#))
[ORA_BIND_OUT](#) ([entier](#))
[ORA_FETCHINTO_ASSOC](#) ([entier](#))
[ORA_FETCHINTO_NULLS](#) ([entier](#))

- [Introduction](#)
- [Installation](#)
- [Constantes pré-définies](#)
- [ora_bind](#)
- [ora_close](#)
- [ora_columnname](#)
- [ora_columnsize](#)
- [ora_columntype](#)
- [ora_commit](#)
- [ora_commitoff](#)
- [ora_commiton](#)
- [ora_do](#)
- [ora_error](#)
- [ora_errorcode](#)
- [ora_exec](#)
- [ora_fetch_into](#)
- [ora_fetch](#)
- [ora_getcolumn](#)
- [ora_logoff](#)
- [ora_logon](#)
- [ora_numcols](#)
- [ora_numrows](#)
- [ora_open](#)
- [ora_parse](#)
- [ora_plogon](#)

- [ora_rollback](#)

8.92.5 [ora_close\(\)](#) : Ferme un pointeur Oracle

bool [ora_close](#) (resource cursor)

[ora_close](#) termine les pointeurs ouverts avec la fonction [ora_open](#) .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. Les erreurs sont accessibles avec les fonctions [ora_error](#) et [ora_errorcode](#) .

8.92.6 [ora_columnname\(\)](#) : Retourne le nom de la colonne de résultat Oracle

string [ora_columnname](#) (resource cursor , int column)

[ora_columnname](#) retourne le nom du champ column du pointeur cursor . Le nom retourné sera en majuscule. La colonne 0 est la première colonne.

8.92.7 [ora_columnsize\(\)](#) : Lit la taille d'une colonne Oracle

int [ora_columnsize](#) (resource cursor , int column)

[ora_columnsize](#) retourne la taille de la colonne column dans le résultat cursor . La colonne 0 est la première colonne.

8.92.8 [ora_columntype\(\)](#) : Retourne le type de la colonne de résultat Oracle

string [ora_columntype](#) (resource cursor , int column)

[ora_columntype](#) retourne le type de la colonne column du résultat cursor . La colonne 0 est la première colonne. Le type retourné prendra une des valeurs suivantes :

- "VARCHAR2"
- "VARCHAR"
- "CHAR"
- "NUMBER"
- "LONG"
- "LONG RAW"
- "ROWID"
- "DATE"
- "CURSOR"

8.92.9 [ora_commit\(\)](#) : Valide une transaction Oracle

bool [ora_commit](#) (resource conn)

ora_commit valide les transactions Oracle en cours sur la connexion Oracle `conn` . Une transaction est définie par toutes les requêtes effectuées sur la connexion `conn` depuis la dernière validation ou annulation (avec auto-validation désactivée) ou depuis l'établissement de la connexion.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.. Les erreurs sont accessibles avec les fonctions ora_error et ora_errorcode .

Voir aussi ora_commiton et ora_commitoff .

8.92.10 ora_commitoff() : Désactive la validation automatique Oracle

bool **ora_commitoff** (resource `conn`)

ora_commitoff désactive la validation automatique après chaque ora_exec , pour la connexion `conn` .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.. Les erreurs sont accessibles avec les fonctions ora_error et ora_errorcode .

Voir aussi ora_commiton et ora_commit .

8.92.11 ora_commiton() : Active la validation automatique Oracle

bool **ora_commiton** (resource `conn`)

ora_commiton active la validation automatique après chaque ora_exec pour la connexion Oracle `conn` .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.. Les erreurs sont accessibles avec les fonctions ora_error et ora_errorcode .

Voir aussi ora_commitoff et ora_commit .

8.92.12 ora_do() : Analyse, exécute et lit le résultat d'une requête Oracle

resource **ora_do** (resource `conn` , string `query`)

ora_do est une combinaison de ora_parse , ora_exec et ora_fetch . Elle va analyser la requête, l'exécuter et lire la première ligne du résultat.

ora_do retourne un index de curseur, ou bien FALSE en cas d'échec. Le détail sur les erreurs est accessible avec les fonctions ora_error et ora_errorcode .

Voir aussi ora_parse , ora_exec et ora_fetch .

8.92.13 ora_error() : Retourne le message d'erreur Oracle

string **ora_error** (resource `cursor_or_connection`)

ora_error retourne un message d'erreur de la forme XXX - NNNNN avec XXX qui est l'origine de l'erreur, et NNNNN qui identifie le message d'erreur.

Note

Le support des connexions a été ajouté en PHP 3.0.4.
--

Avec les versions UNIX d'Oracle, vous pouvez rencontrer des messages d'erreur tels que : \$
Array

00001, 00000,

"unique constraint (%s.%s) violated" // *Cause: An update or insert statement attempted to insert a duplicate key // For Trusted ORACLE configured in DBMS MAC mode, you may see // this message if a duplicate entry exists at a different level. // *Action: Either remove the unique restriction or do not insert the key

8.92.14 ora_errorcode() : Retourne le code d'erreur Oracle

int **ora_errorcode** (*resource cursor_or_connection*)

ora_errorcode retourne le code d'erreur numérique de la dernière commande exécutée sur la connexion ou le pointeur fourni en paramètre.

Note

Les identifiants de connexion ne sont acceptés qu'à partir de la version 3.0.4.

8.92.15 ora_exec() : Exécute une commande analysée sur un pointeur Oracle

bool **ora_exec** (*resource cursor*)

ora_exec exécute la requête préalablement analysée cursor , issue de ora_parse .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.. L'erreur générée sera alors accessible avec les fonctions ora_error et ora_errorcode .

Voir aussi ora_parse , ora_fetch et ora_do .

8.92.16 ora_fetch_into() : Lit une ligne de résultat Oracle dans un tableau

int **ora_fetch_into** (*resource cursor* , *array result* , *int flags*)

ora_fetch_into lit la ligne courante du résultat cursor dans le tableau result . Le paramètre flags peut prendre deux valeurs : s'il vaut ORA_FETCHINTO_NULLS , les colonnes avec comme valeurs NULL seront présentes dans le résultat ; s'il vaut ORA_FETCHINTO_ASSOC , un tableau associatif sera créé.

ora_fetch_into retourne le nombre de colonnes lues.

Exemple avec <u>ora_fetch_into</u>

```

<?php
$results = array();
ora_fetch_into($cursor, $results);
echo $results[0];
echo $results[1];
$results = array();
ora_fetch_into($cursor, $results, ORA_FETCHINTO_NULLS|ORA_FETCHINTO_ASSOC);
echo $results['MyColumn'];
?>

```

Voir aussi [ora_parse](#) , [ora_exec](#) , [ora_fetch](#) et [ora_do](#) .

8.92.17 ora_fetch() : Retourne une ligne de résultat

bool **ora_fetch** (resource cursor)

[ora_fetch](#) lit une ligne de données sur le pointeur cursor .

[ora_fetch](#) retourne TRUE (une ligne a été lue) ou FALSE (plus de lignes à lire ou erreur). Si une erreur survient, [ora_errorcode](#) retournera 0 et la valeur de l'erreur sera disponible dans les fonctions [ora_error](#) et [ora_errorcode](#) .

Voir aussi [ora_parse](#) , [ora_exec](#) et [ora_do](#) .

8.92.18 ora_getcolumn() : Retourne une donnée d'une ligne de résultat Oracle

string **ora_getcolumn** (resource cursor , int column)

[ora_getcolumn](#) lit les données de la colonne column dans le résultat cursor , pour la ligne courante.

[ora_getcolumn](#) retourne la valeur de la colonne. Si une erreur survient, FALSE est retourné et [ora_errorcode](#) aura une valeur non nulle. Notez, qu'un test à FALSE , avec cette fonction peut être \ TRUE , même sans erreur : en effet, la fonction peut retourner des valeurs telles que résultat NULL , chaînes vides, nombre 0, la chaîne "0".

8.92.19 ora_logoff() : Ferme une connexion Oracle

bool **ora_logoff** (resource connection)

[ora_logoff](#) ferme la connexion Oracle connection .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.. Le détail sur les erreurs est accessible avec les fonctions [ora_error](#) et [ora_errorcode](#) .

Voir aussi [ora_logon](#) .

8.92.20 ora_logon() : Ouvre une connexion Oracle

resource **ora_logon** (string user , string password)

[ora_logon](#) établit une connexion entre PHP et un serveur Oracle avec le nom d'utilisateur `user` et le mot de passe `password` .

La connexion peut être faite avec SQL*Net en fournissant le nom TNS de la manière suivante :

Exemple avec [ora_logon](#)

```
<?php
$conn = ora_logon("user@TNSNAME", "pass");
?>
```

Si vous avez des données qui ne sont pas ASCII, vous devriez vérifier que la variable **NLS_LANG** a été correctement configurée dans votre environnement. Pour les modules de serveur, vous devrez la configurer dans l'environnement d'exécution du serveur avant de le lancer.

[ora_logon](#) retourne un index de connexion, en cas de succès, ou FALSE en cas d'échec. Les erreurs sont accessibles avec les fonctions [ora_error](#) et [ora_errorcode](#) .

8.92.21 [ora_numcols\(\)](#) : Retourne le nombre de colonnes

int [ora_numcols](#) (resource cursor)

[ora_numcols](#) retourne le nombre de colonnes dans le résultat `cursor_ind` . Cette valeur n'est significative qu'après une requête `parse/exec/fetch` .

Voir aussi [ora_parse](#) , [ora_exec](#) , [ora_fetch](#) et [ora_do](#) .

8.92.22 [ora_numrows\(\)](#) : Retourne le nombre de lignes

int [ora_numrows](#) (resource cursor)

[ora_numrows](#) retourne le nombre de lignes dans le résultat `cursor` .

8.92.23 [ora_open\(\)](#) : Ouvre un pointeur Oracle

resource [ora_open](#) (resource connection)

[ora_open](#) ouvre un pointeur Oracle sur la connexion.

[ora_open](#) retourne un index de curseur ou bien FALSE en cas d'échec. Les erreurs sont accessibles avec les fonctions [ora_error](#) et [ora_errorcode](#) .

8.92.24 [ora_parse\(\)](#) : Analyse une requête SQL avec Oracle

bool [ora_parse](#) (resource cursor , string sql_statement , int defer)

[ora_parse](#) analyse une requête SQL ou un bloc PL/SQL et l'associe avec le pointeur `cursor` .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

Voir aussi [ora_exec](#) , [ora_fetch](#) et [ora_do](#) .

8.92.25 **ora_plogon()** : Ouvre une connexion persistante à Oracle

resource **ora_plogon** (string user , string password)

ora_plogon établit une connexion persistante à un serveur Oracle, avec l'utilisateur user et le mot de passe password .

Voir aussi ora_logon .

8.92.26 **ora_rollback()** : Annule une transaction Oracle

bool **ora_rollback** (resource connection)

ora_rollback annule une transaction Oracle sur la connexion connection . (Voir ora_commit pour la définition d'une transaction).

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.. Les erreurs sont accessibles avec les fonctions ora_error et ora_errorcode .

8.93 Bufferisation de sortie

8.93.1 Introduction

Les fonctions de bufferisation de sortie vous permettent de contrôler quand les données ont été envoyées par le script. Cela peut être utile dans certaines situations, notamment si vous devez envoyer des en-têtes au navigateur après avoir envoyé des données. Ces fonctions n'affectent pas les en-têtes envoyés par la fonction [header](#) ou les cookies envoyés par [setcookie](#) . Seules les fonctions telles que [echo](#) et les données entre blocs PHP sont affectées.

8.93.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.93.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.93.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable	Historique
<code>output_buffering</code>	"0"	PHP_INI_PERDIR	
<code>output_handler</code>	NULL	PHP_INI_PERDIR	Disponible depuis PHP 4.0.4.
<code>implicit_flush</code>	"0"	PHP_INI_ALL	PHP_INI_PERDIR en PHP <= 4.2.3.

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

`output_buffering` booléen /entier

Vous pouvez activer la bufferisation de sortie pour tous les fichiers avec cette directive, en lui passant la valeur `On` . Si vous souhaitez limiter la taille du buffer à une certaine taille, vous pouvez alors indiquer un nombre maximum d'octets à la place de `On` . Par exemple, `output_buffering=4096`). Depuis PHP 4.3.5, cette directive est toujours désactivée en ligne de commande.

`output_handler` string

Vous pouvez rediriger le résultat de tous vos scripts à une fonction avant leur envoi au navigateur. Par exemple, si vous configurez `output_handler` à [mb_output_handler](#) , l'encodage des caractères sera adapté de manière transparente. Configurer une telle fonction active automatiquement la bufferisation de sortie.

Note

Vous ne pouvez pas utiliser simultanément [mb_output_handler](#) avec [ob_iconv_handler](#) , non plus que [ob_gzhandler](#) avec [zlib.output_compression](#) .

Note

Seules les fonctions internes peuvent être utilisées avec cette directive. Pour les fonctions utilisateurs, utilisez [ob_start](#) .

`implicit_flush` booléen

FALSE par défaut. En changeant cette valeur pour TRUE vous indiquez à PHP que le buffer de sortie doit être vidé automatiquement après chaque fonction d'affichage. Cela revient à appeler la fonction [flush](#) après chaque appel à [print](#) ou [echo](#) et pour tous les blocs HTML .

Lorsque vous utilisez PHP en environnement web, activer cette option a de sérieuses implications et généralement, cela n'est conseillé que pour les débogage. Cette valeur est par défaut à TRUE lorsque PHP fonctionne en mode CLI SAPI .

Voir aussi [ob_implicit_flush](#) .

8.93.5 Types de ressources

Cette extension ne définit aucune ressource.

8.93.6 Constantes pré-définies

Cette extension ne définit aucune constante.

8.93.7 Exemples

Exemple de bufferisation de sortie

```
<?php
ob_start();
echo "Bonjour\n";

setcookie("nom_du_cookie", "valeur_du_cookie");

ob_end_flush();

?>
```

Dans l'exemple ci-dessus, la fonction [echo](#) est stockée dans un buffer jusqu'à l'appel de la fonction [ob_end_flush](#) . Dans le même temps, l'appel à [setcookie](#) a réussi à créer un cookie, sans générer d'erreur. (D'habitude, vous devez envoyer les en-têtes avant les données).

Note

Lorsque vous passez de PHP 4.1 ou 4.2 à 4.3, assurez-vous que `implicit_flush` est à OFF dans votre `php.ini` , sinon la fonction [ob_start](#) ne masquera pas les affichages engendrés.

8.93.8 Voir aussi

Voir aussi [header](#) et [setcookie](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)

- [Exemples](#)
- [Voir aussi](#)
- [flush](#)
- [ob_clean](#)
- [ob_end_clean](#)
- [ob_end_flush](#)
- [ob_flush](#)
- [ob_get_clean](#)
- [ob_get_contents](#)
- [ob_get_flush](#)
- [ob_get_length](#)
- [ob_get_level](#)
- [ob_get_status](#)
- [ob_gzhandler](#)
- [ob_implicit_flush](#)
- [ob_list_handlers](#)
- [ob_start](#)
- [output_add_rewrite_var](#)
- [output_reset_rewrite_vars](#)

8.93.10 [ob_clean\(\)](#) : Efface le buffer de sortie

void [ob_clean](#) (void)

[ob_clean](#) vide le buffer de sortie sans l'envoyer au navigateur.

[ob_clean](#) ne détruit pas le contenu du buffer de sortie comme peut le faire [ob_end_clean](#) .

Voir aussi [ob_flush](#) , [ob_end_flush](#) et [ob_end_clean](#) .

8.93.11 [ob_end_clean\(\)](#) : Détruit les données du buffer de sortie et éteint la bufferisation de sortie

bool [ob_end_clean](#) (void)

[ob_end_clean](#) vide le contenu du premier buffer de sortie et désactive la bufferisation de sortie. Si vous voulez traiter le contenu du buffer, vous devrez appeler [ob_get_contents](#) avant [ob_end_clean](#) , car le buffer est détruit par [ob_end_clean](#) . La fonction TRUE lorsqu'elle a réussi à détruire correctement le buffer, et FALSE sinon. Les raisons d'un tel échec sont que la bufferisation de sortie pouvaient ne pas être activée, ou que, pour une raison quelconque, le buffer n'a pu être détruit.

L'exemple suivant montre comment se débarrasser de tous les buffers de sortie :

Exemple avec ob_end_clean
<pre><?php while (@ob_end_clean()); ?></pre>
Note
Si la fonction échoue, elle génère une note E_NOTICE .

La valeur booléenne retournée par la fonction a été ajoutée en PHP 4.2.0.

Voir aussi [ob_start](#) , [ob_flush](#) et [ob_get_contents](#) .

8.93.12 [ob_end_flush\(\)](#) : Envoie les données du buffer de sortie et éteint la bufferisation de sortie

bool [ob_end_flush](#) (void)

[ob_end_flush](#) envoie le contenu du buffer de sortie (s'il existe) et éteint la bufferisation de sortie. Si vous voulez continuer à manipuler la valeur du buffer, vous pouvez appeler [ob_get_contents](#) avant [ob_end_flush](#) car le contenu du buffer est détruit après un appel à [ob_end_flush](#) . La fonction retourne TRUE lorsqu'elle a réussi à envoyer tout le buffer, sinon FALSE . Les raisons d'échec sont que vous pourriez avoir appelé la fonction sans avoir de buffer actif, pour que pour une raison quelconque, le buffer n'a pu être effacé (possible pour un buffer spécial).

Note

Cette fonction est similaire à [ob_get_flush](#) , excepté que [ob_get_flush](#) retourne le buffer comme une chaîne de caractères.

L'exemple ci-dessous montre une méthode simple pour vider tous les buffers :

Exemple avec [ob_end_flush](#)

```
<?php
while (@ob_end_flush());
?>
```

Note

Si la fonction échoue, elle émet une note E_NOTICE .

Le booléen retourné a été ajouté en PHP 4.2.0.

Voir aussi [ob_start](#) , [ob_get_contents](#) , [ob_get_flush](#) , [ob_flush](#) et [ob_end_clean](#) .

8.93.13 [ob_flush\(\)](#) : Envoie le buffer de sortie

void [ob_flush](#) (void)

[ob_flush](#) envoie le contenu du buffer de sortie (s'il y en a un). Si vous voulez contrôler le contenu du buffer, vous devez appeler la fonction [ob_get_contents](#) avant [ob_flush](#) car le contenu du buffer est effacé après l'appel de [ob_flush](#) .

[ob_flush](#) ne détruit pas le contenu du buffer de sortie comme peut le faire [ob_end_flush](#) .

Voir aussi [ob_get_contents](#) , [ob_clean](#) , [ob_end_flush](#) et [ob_end_clean](#) .

8.93.14 [ob_get_clean\(\)](#) : Lit le contenu courant du buffer de sortie puis l'efface

string [ob_get_clean](#) (void)

[ob_get_clean](#) va retourner le contenu du buffer de sortie, et terminer la session de bufferisation. Si

8.93.11 [ob_end_clean\(\)](#) : Détruit les données du buffer de sortie et éteint la bufferisation de sortie

la bufferisation n'est pas activée, alors `ob_get_clean` retourne FALSE . `ob_get_clean` exécute successivement `ob_get_contents` et `ob_end_clean` .

Exemple avec `ob_get_clean`

```
<?php
ob_start();

print "Bonjour le monde";

$out = ob_get_clean();
$out = strtolower($out);

var_dump($out);
?>
```

L'exemple ci-dessus va afficher :

```
string(16) "Bonjour le monde"
```

```
?>
```

Voir aussi `ob_start` et `ob_get_contents` .

8.93.15 `ob_get_contents()` : Retourne le contenu du buffer de sortie

string `ob_get_contents` (void)

`ob_get_contents` retourne le contenu du buffer de sortie si la bufferisation est activée, ou FALSE sinon.

Voir aussi `ob_start` et `ob_get_length` .

8.93.16 `ob_get_flush()` : Vide le tampon, le retourne en tant que chaîne et stoppe la mise en tampon

string `ob_get_flush` (void)

`ob_get_flush` vide le tampon, le retourne en tant que chaîne et stoppe la mise en tampon.

`ob_get_flush` retourne FALSE si la mise en tampon n'est pas active.

Note

`ob_get_flush` est similaire à `ob_end_flush` , sauf que cette fonction retourne le tampon en tant que chaîne.

Exemple avec `ob_get_flush`

```
<?php
// avec output_buffering=On
print_r(ob_list_handlers());

// sauvegarde le tampon dans un fichier
$buffer = ob_get_flush();
file_put_contents('buffer.txt', $buffer);
```

```
print_r(ob_list_handlers());  
?>
```

L'exemple ci-dessus va afficher :

```
Array  
(  
[0] => default output handler  
)  
Array  
(  
)
```

Voir aussi [ob_end_clean](#) , [ob_end_flush](#) et [ob_list_handlers](#) .

8.93.17 **ob_get_length()** : Retourne la longueur du contenu du buffer de sortie

int **ob_get_length** (void)

[ob_get_length](#) retourne la longueur du contenu du buffer de sortie si la bufferisation est activée, et FALSE sinon.

Voir aussi [ob_start](#) et [ob_get_contents](#) .

8.93.18 **ob_get_level()** : Retourne le nombre de niveaux d'imbrications du système de bufferisation de sortie

int **ob_get_level** (void)

[ob_get_level](#) retourne le nombre de niveaux d'imbrications du système de bufferisation de sortie.

Voir aussi [ob_start](#) et [ob_get_contents](#) .

8.93.19 **ob_get_status()** : Lit le statut du buffer de sortie

array **ob_get_status** (*bool full_status*)

[ob_get_status](#) retourne le statut actuel du système de bufferisation de sortie : un tableau qui contient le statut du buffer, ou bien FALSE en cas d'erreur.

Voir aussi [ob_get_level](#) .

8.93.20 **ob_gzhandler()** : Fonction de callback pour la compression automatique des buffers

string **ob_gzhandler** (string buffer , int mode)

8.93.16 [ob_get_flush\(\)](#) : Vide le tampon, le retourne en tant que chaîne et stoppe la mise en tampon

Note

ob_gzhandler nécessite l'extension zlib .

ob_gzhandler est destinée à être utilisée comme fonction de callback par ob_start pour faciliter l'envoi de données compressées aux navigateurs qui supportent les pages compressées. Avant que ob_gzhandler envoie les données compressées, il détermine les types d'encodage qui sont supportés par le navigateur ("gzip", "deflate" ou aucun) et retourne le contenu des buffers de manière appropriée. Tous les navigateurs sont traités, car c'est aux navigateurs d'envoyer un en-tête indiquant les types de pages supportés.

Note

mode a été ajouté en PHP 4.0.5.

Exemple d'envoi de page compressée avec ob_gzhandler

```
<?php
ob_start("ob_gzhandler");

?>
<html>
  <body>
    <p>Ceci devrait être une page compressée.</p>
  </body>
</html>
```

Note

Vous ne pouvez pas utiliser simultanément ob_gzhandler et zlib.output_compression . De plus, notez bien que zlib.output_compression est préférable à ob_gzhandler .

Voir aussi ob_start et ob_end_flush .

8.93.21 ob_implicit_flush() : Active/désactive l'envoi implicite

```
void ob_implicit_flush ( int flag )
```

ob_implicit_flush active/désactive l'envoi implicite (si flag est fourni. Par défaut, il est activé). L'envoi implicite signifie que toute fonction qui envoie des données au navigateur verra ses données envoyées immédiatement (la fonction flush est appelée automatiquement).

Une fois que l'envoi implicite est désactivé, le buffer de sortie ne sera envoyé qu'au moment de l'appel de ob_end_flush .

Voir aussi flush , ob_start et ob_end_flush .

8.93.22 ob_list_handlers() : Liste les gestionnaires d'affichage utilisés

```
array ob_list_handlers ( void )
```

ob_list_handlers retourne un tableau avec le gestionnaire d'affichage en cours d'utilisation (s'il existe). Si output_buffering est activé ou si une fonction anonyme est utilisée avec ob_start , ob_list_handlers retournera un tableau avec comme valeur d'entrée : "default output handler".

Exemple avec ob_list_handlers

```
<?php
```

```
// avec output_buffering=On
print_r(ob_list_handlers());
ob_end_flush();

ob_start("ob_gzhandler");
print_r(ob_list_handlers());
ob_end_flush();

// fonctions anonymes
ob_start(create_function('$string', 'return $string;'));
print_r(ob_list_handlers());
ob_end_flush();
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[0] => default output handler
)

Array
(
[0] => ob_gzhandler
)

Array
(
[0] => default output handler
)
```

Voir aussi [ob_end_clean](#) , [ob_end_flush](#) , [ob_get_flush](#) et [ob_start](#) .

8.93.23 ob_start() : Enclenche la bufferisation de sortie

`bool ob_start (callback output_callback , int chunk_size , bool erase)`

[ob_start](#) démarre la bufferisation de sortie. Tant qu'elle est enclenchée, aucune donnée, hormis les en-têtes, n'est envoyée au navigateur, mais temporairement mise en buffer.

Le contenu de ce buffer peut être copié dans une chaîne avec la fonction [ob_get_contents](#) . Pour afficher le contenu de ce buffer, utilisez [ob_end_flush](#) . = Au contraire, [ob_end_clean](#) effacera le contenu de ce buffer.

Une fonction optionnelle de callback peut être spécifiée en troisième argument. [ob_start](#) prend une chaîne comme paramètre, et retourne une chaîne. Elle sera appelée par [ob_end_flush](#) ou lorsque le buffer sera envoyé au navigateur à la fin du script et recevra le contenu du buffer de sortie. Lorsque la fonction `output_callback` est appelée, elle doit retourner un nouveau contenu pour le buffer de sortie : celui-ci sera envoyé au navigateur. Si `output_callback` n'est pas une fonction accessible, la fonction retournera `FALSE` . Si la fonction de callback a deux paramètres, le second est composé du champ bits constitué par `PHP_OUTPUT_HANDLER_START` , `PHP_OUTPUT_HANDLER_CONT` et `PHP_OUTPUT_HANDLER_END` .

Note

En PHP 4.0.4, `ob_gzhandler` a été introduite pour faciliter l'envoi de fichiers compressés avec gz aux navigateurs web qui supportent les pages compressées. `ob_gzhandler` détermine le type d'encodage accepté par un navigateur, et retourne le contenu le plus adéquat.

Note

Avant PHP 4.3.2, cette fonction ne renvoyait pas FALSE dans le cas où `output_callback` ne pouvait être exécutée.

Attention

Quelques serveurs web (e.g. Apache) modifient le dossier de travail d'un script lorsqu'il appelle une fonction de callback. Vous pouvez revenir à un comportement normal en ajoutant `chdir(dirname($_SERVER['SCRIPT_FILENAME']))` dans votre fonction de callback.

Si le paramètre optionnel `chunk_size` est passé, la fonction de callback est appelée à chaque nouvelle ligne après `chunk_size` octets d'affichage. Le paramètre `output_callback` peut être outrepassé en lui affectant la valeur NULL .

Si le paramètre optionnel `erase` est défini à FALSE , le buffer ne sera pas effacé tant que le script ne sera pas terminé (depuis PHP 4.3.0).

Les buffers de sortie sont gérés par pile, c'est-à-dire que vous pouvez appeler plusieurs fois `ob_start` simultanément. Assurez-vous que vous appelez `ob_end_flush` suffisamment souvent. Si plusieurs fonctions de callback sont actives, les contenus seront filtrés séquentiellement, dans l'ordre d'emboîtement.

`ob_end_clean` , `ob_end_flush` , `ob_clean` , `ob_flush` et `ob_start` ne doivent pas être appelés depuis une fonction de callback. Si vous les appelez depuis une fonction de callback, le comportement ne sera pas défini. Si vous voulez effacer le contenu du buffer, retournez "" (une chaîne vide) comme fonction de callback. Vous ne pourrez jamais appeler les fonctions utilisant la fonction de bufferisation de sortie comme `print_r($expression, true)` ou `highlight_file($filename, true)` depuis une fonction de callback.

Exemple de gestion de sortie avec fonction utilisateur

```
<?php
function callback($buffer)
{
    // remplace toutes les pommes par des carottes
    return (ereg_replace("pommes de terre", "carottes", $buffer));
}

ob_start("callback");

?>

<html>
<body>
<p>C'est comme comparer des carottes et des pommes de terre.</p>
</body>
</html>

<?php
ob_end_flush();

?>
```

L'exemple ci-dessus va afficher :

```
<html>
```

```
<body>
  <p>C'est comme comparer des carottes et des carottes.</p>
</body>
</html>
```

Voir aussi [ob_get_contents](#) , [ob_end_flush](#) , [ob_end_clean](#) , [ob_implicit_flush](#) , [ob_gzhandler](#) , [ob_iconv_handler](#) , [mb_output_handler](#) et [ob_tidyhandler](#) .

8.93.24 [output_add_rewrite_var\(\)](#) : Ajoute une règle de réécriture d'URL

bool [output_add_rewrite_var](#) (string name , string value)

[output_add_rewrite_var](#) réécrit les URLs avec la variable donnée.

Note

Cette fonction met en tampon la sortie.

Exemple avec [output_add_rewrite_var](#)

```
<?php
output_add_rewrite_var('var', 'value');

// un lien
echo '<a href="file.php">link</a>';

// un formulaire
echo '<form action="script.php" method="post">
<input type="text" name="var2" />
</form>';

print_r(ob_list_handlers());
?>
```

L'exemple ci-dessus va afficher :

```
<a href="file.php?var=value">link</a>

<form action="script.php" method="post">
<input type="hidden" name="var" value="value" />
<input type="text" name="var2" />
</form>
```

Array

```
(
[0] => URL-Rewriter
)
```

Voir aussi [output_reset_rewrite_vars](#) , [ob_flush](#) et [ob_list_handlers](#) .

8.93.25 [output_reset_rewrite_vars\(\)](#) : Annule la réécriture d'URL

bool [output_reset_rewrite_vars](#) (void)

[output_reset_rewrite_vars](#) efface la règle de réécriture d'URLs et annule les changements faits par [output_add_rewrite_var](#) et/ou par [session_start](#) et qui sont encore en tampon.

Exemple avec `output_reset_rewrite_vars`

```
<?php
session_start();
output_add_rewrite_var('var', 'value');

echo '<a href="file.php">link</a>';
ob_flush();

output_reset_rewrite_vars();
echo '<a href="file.php">link</a>';
?>
```

L'exemple ci-dessus va afficher :

```
<a href="file.php?PHPSESSID=xxx&var=value">link</a>
<a href="file.php">link</a>
```

Voir aussi `output_add_rewrite_var` , `ob_flush` , `ob_list_handlers` et `session_start` .

8.94 Overload

8.94.1 Introduction

Le but de cette extension est de permettre de maîtriser les appels aux méthodes et aux membres d'un objet. Seule une fonction est définie dans cette extension, `overload` qui demande le nom de la classe qui doit supporter cette fonctionnalité. Cette classe doit être pourvue des méthodes nécessaires au bon fonctionnement de l'extension, c'est-à-dire : `__get()` , `__set()` et `__call()` , qui servent respectivement à lire, à modifier un membre et à appeler une méthode. De cette manière, l'overloading assure un contrôle sur les fonctions appelées. A l'intérieur de ces méthodes, l'overloading est désactivé, pour que vous puissiez accéder à l'objet.

Attention

Ce module est EXPERIMENTAL . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez ce module à vos risques et périls.

Attention

Cette extension ne fait pas partie de PHP 5. PHP 5 supporte <code>__get()</code> , <code>__set()</code> et <code>__call()</code> nativement. Voir la page traitant de la surcharge en PHP 5 pour plus d'informations.

8.94.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.94.3 Installation

Afin d'utiliser ces fonctions, vous devez compiler PHP avec l'option `--enable-overload` . Depuis PHP 4.3.0, cette extension est activée par défaut. Pour la désactiver, utilisez l'option `--disable-overload` .

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de bibliothèque supplémentaire pour disposer de ces fonctions.

Note

Le support par défaut de l'extension <code>overload</code> a été ajouté en PHP 4.3.0.

8.94.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.94.5 Types de ressources

Cette extension ne définit aucune ressource.

8.94.6 Constantes pré-définies

Cette extension ne définit aucune constante.

8.94.7 Exemples

Voici un exemple simple de fonctions utilisant overload :

Overload avec une classe PHP

```
<?php

class OO {
    var $a = 111;
    var $elem = array('b' => 9, 'c' => 42);

    // Fonction de callback pour la lecture de membre
    function __get($prop_name, &$prop_value)
    {
        if (isset($this->elem[$prop_name])) {
            $prop_value = $this->elem[$prop_name];
            return true;
        } else {
            return false;
        }
    }

    // Fonction de callback pour l'écriture de membre
    function __set($prop_name, $prop_value)
    {
        $this->elem[$prop_name] = $prop_value;
        return true;
    }
}

// Ici, l'initiation de l'overload
overload('OO');

$o = new OO;
echo "\$o->a: $o->a\n"; // $o->a: 111
echo "\$o->b: $o->b\n"; // $o->b: 9
echo "\$o->c: $o->c\n"; // $o->c: 42
echo "\$o->d: $o->d\n"; // $o->d:

// ajouter une nouvelle valeur au membre $elem, en programmation OOP
$o->x = 56;

// instantiation de la classe stdClass (elle existe par défaut en PHP 4)
// $val n'est pas overloadée!
$val = new stdClass;
$val->prop = 555;

// Forcez "a" à être un tableau avec l'élément $val
// Mais __set() forcera cet élément dans le tableau $elem
$o->a = array($val);
var_dump($o->a[0]->prop);

?>
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [overload](#)

8.95 Ovrimos SQL

8.95.1 Introduction

Ovrimos SQL Server est une base de données relationnelle client/serveur et transactionnelle, combinée avec des fonctionnalités web, et des transactions rapides.

Note
Cette extension a été déplacée dans le module <u>PECL</u> et ne sera plus intégrée dans PHP à partir de PHP 5.1.0.
Note
Cette extension n'est pas disponible sur les plates-formes Windows.

8.95.2 Pré-requis

Vous devez installer la bibliothèque sqlcli, qui est disponible dans la distribution de Ovrimos SQL Server.

8.95.3 Installation

Pour activer le support d'Ovrimos en PHP, vous devez compiler PHP avec l'option `--with-ovrimos[=DIR]`. DIR est le chemin jusqu'au dossier d'Ovrimos qui contient la bibliothèque libsqlcli.

8.95.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.95.5 Types de ressources

8.95.6 Exemples

Connexion au serveur Ovrimos SQL Server et sélection d'une table système
<pre><?php \$conn = ovrimos_connect("server.domain.com", "8001", "admin", "password"); if (\$conn != 0) { echo 'Connexion établie!'; \$res = ovrimos_exec(\$conn, "select table_id, table_name from sys.tables"); if (\$res != 0) { echo 'Requête effectuée!'; ovrimos_result_all(\$res); ovrimos_free_result(\$res); } ovrimos_close(\$conn); } ?></pre>

Cet exemple effectue une connexion à un serveur Ovrimos SQL Server.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Exemples](#)
- [ovrimos_close](#)
- [ovrimos_commit](#)
- [ovrimos_connect](#)
- [ovrimos_cursor](#)
- [ovrimos_exec](#)
- [ovrimos_execute](#)
- [ovrimos_fetch_into](#)
- [ovrimos_fetch_row](#)
- [ovrimos_field_len](#)
- [ovrimos_field_name](#)
- [ovrimos_field_num](#)
- [ovrimos_field_type](#)
- [ovrimos_free_result](#)
- [ovrimos_longreadlen](#)
- [ovrimos_num_fields](#)
- [ovrimos_num_rows](#)
- [ovrimos_prepare](#)
- [ovrimos_result_all](#)
- [ovrimos_result](#)
- [ovrimos_rollback](#)

8.95.8 [ovrimos_commit\(\)](#) : Valide une transaction Ovrimos

bool [ovrimos_commit](#) (int connection_id)

[ovrimos_commit](#) exécute la transaction préparée sur la connexion `connection_id` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.95.9 [ovrimos_connect\(\)](#) : Connexion à un serveur

int [ovrimos_connect](#) (string host , string db , string user , string password)

[ovrimos_connect](#) se connecte à un serveur Ovrimos.

[ovrimos_connect](#) retourne un identifiant de connexion, supérieur à 0, ou 0 en cas d'échec. `host` est l'adresse IP de l'hôte Ovrimos, et `db` est soit le nom d'une base de données, soit une chaîne contenant le numéro de port.

Exemple avec [ovrimos_connect](#)

```
<?php
$conn = ovrimos_connect("server.domain.com", "8001", "admin", "password");
if ($conn != 0) {
    echo 'Connexion établie!';
    $res=ovrimos_exec($conn, "select table_id, table_name from sys.tables");
    if ($res != 0) {
        echo 'Requête effectuée!';
        ovrimos_result_all($res);
    }
}
```

```

        ovrimos_free_result($res);
    }
    ovrimos_close($conn);
}
?>

```

L'exemple ci-dessus montre comment se connecter à une base de données et afficher le contenu d'une table.

8.95.10 ovrimos_cursor() : Retourne le nom du curseur Ovrimos

string **ovrimos_cursor** (int result_id)

ovrimos_cursor retourne le nom du curseur. Pratique, lorsqu'on veut faire des modifications ou des effacements avec des curseurs déjà positionnés.

8.95.11 ovrimos_exec() : Exécute une requête SQL

int **ovrimos_exec** (int connection_id , string query)

ovrimos_exec exécute une requête SQL (sélection ou modification), et retourne un identifiant de résultat result_id (ou bien FALSE , en cas d'échec). Evidemment, la requête SQL ne doit pas contenir de paramètres.

8.95.12 ovrimos_execute() : Exécute une requête Ovrimos préparée

bool **ovrimos_execute** (int result_id , array parameters_array)

ovrimos_execute sert à exécuter une requête SQL. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. Si la requête préparée contient des paramètres (des points d'interrogation dans la requête), un nombre correct de paramètres doit être passé dans le tableau parameters_array . Notez que ovrimos_execute ne suit pas les conventions PHP qui placent les noms des paramètres entre crochets. L'auteur n'a pas pu s'y faire.

8.95.13 ovrimos_fetch_into() : Lit une ligne dans un résultat Ovrimos

bool **ovrimos_fetch_into** (int result_id , array result_array , string how , int rownumber)

ovrimos_fetch_into lit une ligne dans le résultat result_id , qui doit être passé en référence. La ligne qui sera lue est déterminée par les deux paramètres how et rownumber . how peut prendre les valeurs de Next (suivant, valeur par défaut), Prev (précédent), First (premier), Last (dernier), Absolute (position absolue). La casse de how n'est pas prise en compte. rownumber est optionnel, sauf dans le cas d' Absolute . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Lit un exemple

```

<?php
$conn=ovrimos_connect("neptune", "8001", "admin", "password");
if ($conn!=0) {
    echo 'Connexion établie!';
    $res=ovrimos_exec($conn, "SELECT table_id, table_name FROM sys.tables");
}
?>

```

```

if ($res != 0) {
    echo 'Requête effectuée!';
    if (ovrimos_fetch_into($res, &$row)) {
        list ($table_id, $table_name) = $row;
        echo "table_id=".$table_id.", table_name=".$table_name."\n";
        if (ovrimos_fetch_into($res, &$row)) {
            list ($table_id, $table_name) = $row;
            echo "table_id=".$table_id.", table_name=".$table_name."\n";
        } else {
            echo "Next: erreur\n";
        }
    } else {
        echo "First: erreur\n";
    }
    ovrimos_free_result($res);
}
ovrimos_close($conn);
}
?>

```

Cet exemple lit une ligne.

8.95.14 `ovrimos_fetch_row()` : Lit une ligne dans un résultat Ovrimos

`bool ovrimos_fetch_row (int result_id , int how , int row_number)`

`ovrimos_fetch_row` lit une ligne dans un résultat. Les colonnes doivent être lues par un autre appel. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple de lecture de ligne Ovrimos

```

<?php
$conn = ovrimos_connect("remote.host", "8001", "admin", "password");
if ($conn != 0) {
    echo 'Connexion établie!';
    $res=ovrimos_exec($conn, "select table_id, table_name from sys.tables");
    if ($res != 0) {
        echo 'Requête effectuée!';
        if (ovrimos_fetch_row($res, "First")) {
            $table_id = ovrimos_result($res, 1);
            $table_name = ovrimos_result($res, 2);
            echo "table_id=".$table_id.", table_name=".$table_name."\n";
            if (ovrimos_fetch_row($res, "Next")) {
                $table_id = ovrimos_result($res, "table_id");
                $table_name = ovrimos_result($res, "table_name");
                echo "table_id=".$table_id.", table_name=".$table_name."\n";
            } else {
                echo "Next: erreur\n";
            }
        } else {
            echo "First: erreur\n";
        }
        ovrimos_free_result($res);
    }
    ovrimos_close($conn);
}
?>

```

Cet exemple lit une ligne et l'affiche.

8.95.15 ovrimos_field_len() : Retourne la taille d'une colonne Ovrimos

int **ovrimos_field_len** (int result_id , int field_number)

ovrimos_field_len retourne la taille de la colonne numéro field_number , dans le résultat result_id .

8.95.16 ovrimos_field_name() : Retourne le nom d'une colonne Ovrimos

string **ovrimos_field_name** (int result_id , int field_number)

ovrimos_field_name retourne le nom d'une colonne à partir de son numéro de colonne field_number , (la première colonne est à 1).

8.95.17 ovrimos_field_num() : Retourne le numéro de colonne Ovrimos

int **ovrimos_field_num** (int result_id , string field_name)

ovrimos_field_num retourne le numéro de la colonne field_name (la numérotation commence à 1), dans result_id ou FALSE en cas d'erreur.

8.95.18 ovrimos_field_type() : Retourne le type numérique d'une colonne Ovrimos

int **ovrimos_field_type** (int result_id , int field_number)

ovrimos_field_type retourne le type numérique d'une colonne, identifiée par son numéro field_number dans le résultat result_id .

8.95.19 ovrimos_free_result() : Libère les ressources utilisées par un résultat Ovrimos

bool **ovrimos_free_result** (int result_id)

ovrimos_free_result libère toutes les ressources prises par le résultat result_id . ovrimos_free_result retourne TRUE .

8.95.20 ovrimos_longreadlen() : Indique la taille des données à lire dans une colonne de grande taille

bool **ovrimos_longreadlen** (int result_id , int length)

ovrimos_longreadlen indique le nombre d'octets qui seront lus dans une colonne de grande taille (long varchar et long varbinary). Par défaut, 0. ovrimos_longreadlen assigne cette valeur au résultat courant. ovrimos_longreadlen retourne TRUE .

8.95.21 `ovrimos_num_fields()` : Retourne le nombre de colonnes Ovrimos

int `ovrimos_num_fields` (int `result_id`)

`ovrimos_num_fields` indique le nombre de colonnes du résultat `result_id` .

8.95.22 `ovrimos_num_rows()` : Retourne le nombre de lignes affectées par une modification Ovrimos

int `ovrimos_num_rows` (int `result_id`)

`ovrimos_num_rows` retourne le nombre de lignes affectées par une requête SQL d'UPDATE.

8.95.23 `ovrimos_prepare()` : Prépare une requête SQL Ovrimos

int `ovrimos_prepare` (int `connection_id` , string `query`)

`ovrimos_prepare` prépare une requête SQL et retourne un identifiant de résultat (ou FALSE en cas d'échec).

Connexion à un serveur Ovrimos SQL Server et préparation d'une requête

```
<?php
$conn=ovrimos_connect("db_host", "8001", "admin", "password");
if ($conn!=0) {
    echo 'Connexion établie!';
    $res=ovrimos_prepare($conn, "select table_id, table_name
                                from sys.tables where table_id=1");

    if ($res != 0) {
        echo 'Préparation faite!';
        if (ovrimos_execute($res)) {
            echo "Exécution réussie!\n";
            ovrimos_result_all($res);
        } else {
            echo 'Exécution manquée!';
        }
        ovrimos_free_result($res);
    } else {
        echo "Préparation manquée!\n";
    }
    ovrimos_close($conn);
}
?>
```

Cet exemple montre comment se connecter à un serveur Ovrimos SQL Server, comment préparer une requête SQL et l'exécuter.

8.95.24 `ovrimos_result_all()` : Affiche un résultat Ovrimos sous forme de table HTML

int `ovrimos_result_all` (int `result_id` , string *format*)

`ovrimos_result_all` affiche le résultat de la requête représentée par `result_id`. `ovrimos_result_all` retourne le nombre de lignes générées.

Prépare une requête, l'exécute sur Ovrimos et affiche le résultat

```
<?php
$conn = ovrimos_connect("db_host", "8001", "admin", "password");
if ($conn != 0) {
    echo 'Connexion établie!';
    $res = ovrimos_prepare($conn, "select table_id, table_name
                                from sys.tables where table_id = 7");

    if ($res != 0) {
        echo 'Préparation faite!';
        if (ovrimos_execute($res, array(3))) {
            echo "Exécution réussie!\n";
            ovrimos_result_all($res);
        } else {
            echo 'Exécution manquée!';
        }
        ovrimos_free_result($res);
    } else {
        echo "Préparation manquée!\n";
    }
    ovrimos_close($conn);
}
?>
```

Cet exemple exécute une requête SQL et affiche le résultat sous forme d'une table HTML.

`ovrimos_result_all` avec meta-information

```
<?php
$conn = ovrimos_connect("db_host", "8001", "admin", "password");
if ($conn != 0) {
    echo 'Connexion établie!';
    $res = ovrimos_exec($conn, "select table_id, table_name
                                from sys.tables where table_id = 1");

    if ($res != 0) {
        echo "Requête effectuée! cursor=" . ovrimos_cursor($res) . "\n";
        $colnb = ovrimos_num_fields($res);
        echo "Output columns=" . $colnb . "\n";
        for ($i=1; $i<=$colnb; $i++) {
            $name = ovrimos_field_name($res, $i);
            $type = ovrimos_field_type($res, $i);
            $len = ovrimos_field_len($res, $i);
            echo "Colonne ".$i." nom=".$name." type=".$type." longueur=".$len."\n";
        }
        ovrimos_result_all($res);
        ovrimos_free_result($res);
    }
    ovrimos_close($conn);
}
?>
```

Exemple avec `ovrimos_result_all`

```
<?php
$conn = ovrimos_connect("db_host", "8001", "admin", "password");
if ($conn != 0) {
    echo 'Connexion établie!';
    $res = ovrimos_exec($conn, "update test set i=5");
    if ($res != 0) {
        echo 'Requête effectuée!';
        echo ovrimos_num_rows($res) . " lignes affectées\n";
        ovrimos_free_result($res);
    }
}
```

```
}  
    ovrimos_close($conn);  
}  
?>
```

8.95.25 ovrimos_result() : Lit le contenu d'une colonne Ovrimos

string **ovrimos_result** (int result_id , mixed field)

ovrimos_result lit le contenu de la colonne field dans le résultat result_id . field peut être le nom de la colonne (une chaîne) ou bien le numéro de la colonne (la première colonne est alors 1).

ovrimos_result retourne FALSE en cas d'échec.

8.95.26 ovrimos_rollback() : Annule une transaction

bool **ovrimos_rollback** (int connection_id)

ovrimos_rollback annule la transaction préparée sur la connexion connection_id . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.96 Accès aux fichiers Paradox

8.96.1 Introduction

Attention
Ce module est EXPERIMENTAL . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez ce module à vos risques et périls.

Ce module vous permet de lire des bases de données Paradox ainsi que des fichiers d'index primaires. Il peut également créer des bases de données Paradox. Considérez cette fonctionnalité comme expérimentale à cause d'un manque de documentation du format de fichier Paradox.

Note
Ce module n'a pas été testé sur d'autre plateforme que Debian/GNU Linux pour PPC.
Note
Ce module est également toujours en cours de développement et peut changer, même si l'API ne sera surement pas modifié en profondeur.

8.96.2 Pré-requis

Vous devez avoir au minimum PHP 5.0.0 et libpx >= 0.1.9. La bibliothèque Paradox (libpx) est disponible sur <http://pxlib.sourceforge.net> .

8.96.3 Installation

Une courte note d'installation : entrez juste
`$ pear install paradox`
dans votre console.

8.96.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.96.5 API orientée objet

L'extension Paradox fournit également une API orientée objet. Elle consiste en une seule classe nommée `paradox_doc`. Ses méthodes diffèrent uniquement des fonctions en leurs noms et, bien sûr, le premier paramètre manquant. La table suivante liste toutes les méthodes avec les fonctions équivalentes.

Nom de la méthode	Fonction équivalente
Constructor	<code>px_new</code>
Destructor	<code>px_delete</code>
<code>open_fp</code>	<code>px_open_fp</code>
<code>create_fp</code>	<code>px_create_fp</code>
<code>close</code>	<code>px_close</code>

<u>numrecords</u>	<u>px_numrecords</u>
<u>numfields</u>	<u>px_numfields</u>
<u>get_record</u>	<u>px_get_record</u>
<u>put_record</u>	<u>px_put_record</u>
<u>get_field</u>	<u>px_get_field</u>
<u>get_schema</u>	<u>px_get_schema</u>
<u>get_info</u>	<u>px_get_info</u>
<u>set_parameter</u>	<u>px_set_parameter</u>
<u>get_parameter</u>	<u>px_get_parameter</u>
<u>set_value</u>	<u>px_set_value</u>
<u>get_value</u>	<u>px_get_value</u>
<u>get_info</u>	<u>px_get_info</u>
<u>set_targetencoding</u>	<u>px_set_targetencoding</u>
<u>set_tablename</u>	<u>px_set_tablename</u>
<u>set_blob_file</u>	<u>px_set_blob_file</u>
<u>timestamp2string</u>	<u>px_timestamp2string</u>

8.96.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Les deux tables suivantes listent toutes les constantes définies par l'extension Paradox.

Nom	Signification
PX_FIELD_ALPHA	Données sous forme de caractères avec une longueur fixe
PX_FIELD_DATE	Date
PX_FIELD_SHORT	Entier court (2 octets)
PX_FIELD_LONG	Entier long (4 octets)
PX_FIELD_CURRENCY	Identique à PX_FIELD_NUMBER
PX_FIELD_NUMBER	Double
PX_FIELD_LOGICAL	Booléen
PX_FIELD_MEMOBLOB	Objet binaire large (non supporté)
PX_FIELD_BLOB	Objet binaire large (non supporté)
PX_FIELD_FMTMEMOBLOB	Objet binaire large (non supporté)
PX_FIELD_OLE	Objet OLE (basiquement un blob, non supporté)
PX_FIELD_GRAPHIC	Graphique (basiquement un blob, non supporté)
PX_FIELD_TIME	time
PX_FIELD_TIMESTAMP	timestamp (comme un timestamp Unix)
PX_FIELD_AUTOINC	Entier auto-incrémenté (comme PX_FIELD_LONG)
PX_FIELD_BCD	Nombre décimal stocké en format bcd (non supporté)
PX_FIELD_BYTES	Tableau d'octets d'une longueur maximale de 255 octets (non supporté)
PX_KEYTOLOWER	
PX_KEYTOUPPER	
Nom	Signification

PX_FILE_INDEX_DB	Base de données indexée
PX_FILE_PRIM_INDEX	Index primaire
PX_FILE_NON_INDEX_DB	Base de données non indexée
PX_FILE_NON_INC_SEC_INDEX	Non-indexé incrémentalement secondaire (non supporté)
PX_FILE_SEC_INDEX	Index secondaire (non supporté)
PX_FILE_INC_SEC_INDEX	Index secondaire incrémental (non supporté)
PX_FILE_NON_INC_SEC_INDEX_G	(non supporté)
PX_FILE_SEC_INDEX_G	(non supporté)
PX_FILE_INC_SEC_INDEX_G	(non supporté)

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [API orientée objet](#)
- [Constantes pré-définies](#)
- [px_close](#)
- [px_create_fp](#)
- [px_delete](#)
- [px_get_field](#)
- [px_get_info](#)
- [px_get_parameter](#)
- [px_get_record](#)
- [px_get_schema](#)
- [px_get_value](#)
- [px_new](#)
- [px_numfields](#)
- [px_numrecords](#)
- [px_open_fp](#)
- [px_put_record](#)
- [px_set_blob_file](#)
- [px_set_parameter](#)
- [px_set_tablename](#)
- [px_set_targetencoding](#)
- [px_set_value](#)
- [px_timestamp2string](#)

8.96.8 [px_create_fp\(\)](#) : Crée une nouvelle base de données Paradox

bool [px_create_fp](#) (resource pxdoc , resource file , array fielddesc)

Crée un nouveau fichier de base de données Paradox. Le fichier actuel doit avoir été ouvert avec la fonction [fopen](#) . Assurez-vous que ce fichier est accessible en écriture. Le premier paramètre est la valeur retournée par la fonction [px_new](#) . fielddesc est un tableau comprenant un élément pour chaque champ et contenant les spécifications du champ. Chaque spécification de champ est un tableau avec deux ou trois éléments. Le premier élément est toujours une chaîne de caractères représentant le nom du champ. Il ne peut être plus grand que dix caractères. Le deuxième élément contient le type du champ qui peut être une des constantes listées dans la table " [Constantes pour les types de champ](#) ". Dans le cas d'un champ sous la forme de caractères, vous devez fournir un troisième élément spécifiant la longueur du champ.

Note

Cette fonction est hautement expérimentale, à cause du manque de documentation du format de fichier Paradox. Les fichiers de base de données créés avec cette fonction peuvent être ouvert avec la fonction `px_open_fp` et devraient s'ouvrir sans problème avec les logiciels Paradox.

8.96.9 px_delete() : Efface la ressource d'une base de données Paradox

bool `px_delete` (resource pxdoc)

Efface la ressource d'une base de données Paradox et libère toute la mémoire.

8.96.10 px_get_field() : Retourne les spécifications d'un champ

array `px_get_field` (resource pxdoc , int fieldno)

8.96.11 px_get_info() : Retourne des informations sur un fichier Paradox

array `px_get_info` (resource pxdoc)

8.96.12 px_get_parameter() : Récupère un paramètre

string `px_get_parameter` (resource pxdoc , string name)

Récupère divers paramètres.

8.96.13 px_get_record() : Retourne un enregistrement d'une base de données Paradox

array `px_get_record` (resource pxdoc , int num , int mode)

8.96.14 px_get_schema() : Retourne le schéma de la base de données

array `px_get_schema` (resource pxdoc , int mode)

`px_get_schema` retourne le schéma de la base de données.

8.96.15 px_get_value() : Récupère une valeur

float `px_get_value` (resource pxdoc , string name)

Récupère diverses valeurs.

8.96.16 `px_new()` : Crée un nouvel objet Paradox

resource `px_new` ()

Crée un nouvel objet Paradox. Vous devez appeler cette fonction avant toute autre fonction. `px_new` ne crée aucun fichier sur le disque, il crée juste une instance d'un objet Paradox.

8.96.17 `px_numfields()` : Retourne le nombre de champs d'une base de données Paradox

int `px_numfields` (resource `pxdoc`)

Récupère le nombre de champs dans un fichier de base de données Paradox.

8.96.18 `px_numrecords()` : Retourne le nombre d'enregistrements d'une base de données Paradox

int `px_numrecords` (resource `pxdoc`)

Retourne le nombre d'enregistrements d'un fichier de base de données Paradox.

8.96.19 `px_open_fp()` : Ouvre une base de données Paradox

bool `px_open_fp` (resource `pxdoc` , resource `file`)

Ouvre un fichier de base de données Paradox existant. Le fichier actuel doit avoir été ouvert avant par la fonction `fopen` . Cette fonction peut également être utilisée pour ouvrir les fichiers d'index primaires et les traiter comme des bases de données Paradox. Ceci est supporté pour ceux qui veulent investiguer un index primaire. Il ne peut être utilisé pour accélérer l'accès à un fichier de base de données.

8.96.20 `px_put_record()` : Stock un enregistrement dans une base de données Paradox

bool `px_put_record` (resource `pxdoc` , array `record`)

Stock un enregistrement dans une base de données Paradox. L'enregistrement sera ajouté à la fin de la base de données.

8.96.21 `px_set_blob_file()` : Définit le fichier depuis lequel les blobs seront lus

bool `px_set_blob_file` (resource `pxdoc` , string `filename`)

Définit le nom du fichier depuis lequel les blobs seront lus. Sans l'appel à cette fonction, `px_get_record` retournera uniquement les données des champs blob si les données sont des parties des enregistrements et ne sont pas stockées dans un fichier blob. Les données blob sont

stockées dans un enregistrement si elles sont suffisamment petites pour correspondre à la taille du champ blob.

Deux appels à cette fonction fermera le premier fichier blob et ouvrira le nouveau.

8.96.22 `px_set_parameter()` : Définit un paramètre

`bool px_set_parameter (resource pxdoc , string name , string value)`

Définit divers paramètres.

8.96.23 `px_set_tablename()` : Définit le nom de la table (déprécié)

`void px_set_tablename (resource pxdoc , string name)`

Définit le nom de la table d'une base de données Paradox, qui a été créé avec la fonction `px_create_fp`. Appliquer cette fonction sur une base de données existante n'a aucun effet.

8.96.24 `px_set_targetencoding()` : Définit l'encodage pour les caractères des champs (déprécié)

`bool px_set_targetencoding (resource pxdoc , string encoding)`

Définit l'encodage pour les données récupérées depuis les champs. Tous les caractères des champs seront réencodés par cette fonction. Si l'encodage n'est pas défini, les caractères des données seront retournés en utilisant l'encodage de page DOS comme spécifié dans la base de données.

Voir aussi `px_get_info` pour déterminer la page d'encodage DOS.

8.96.25 `px_set_value()` : Définit une valeur

`bool px_set_value (resource pxdoc , string name , float value)`

Définit diverses valeurs.

8.96.26 `px_timestamp2string()` : Convertit un timestamp en une chaîne de caractères

`string px_timestamp2string (resource pxdoc , float value , string format)`

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.97 Fonctions Parsekit

8.97.1 Introduction

Ces fonctions vous permettent d'analyser en cours de fonctionnement l'opcode compilé depuis vos scripts PHP.

8.97.2 Installation

Cette extension PECL n'est pas intégrée à PHP.

Des informations comme les notes sur les nouvelles versions, les téléchargements, les sources des fichiers, les informations concernant les mainteneurs ainsi qu'un CHANGELOG, peuvent être trouvées ici : <http://pecl.php.net/package/parsekit> .

Vous pouvez télécharger ces bibliothèques DLL pour ces extensions PECL depuis la page de téléchargement de PHP sous sur la page <http://snaps.php.net/> .

8.97.3 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.97.4 Types de ressources

Cette extension ne définit aucune ressource.

8.97.5 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

PARSEKIT_QUIET (entier)

Retourne tous les détails mais sans les entrées NULL non nécessaire.

PARSEKIT_SIMPLE (entier)

Retourne une notation opcode abrégée.

PARSEKIT_EXTENDED_VALUE (entier)

Flag d'un noeud opcode

PARSEKIT_RESULT_CONST (entier)

Flag d'un noeud opcode

PARSEKIT_RESULT_EA_TYPE (entier)

Flag d'un noeud opcode

PARSEKIT_RESULT_JMP_ADDR (entier)

Flag d'un noeud opcode

PARSEKIT_RESULT_OPARRAY (entier)

Flag d'un noeud opcode

PARSEKIT_RESULT_OPLINE (entier)

Flag d'un noeud opcode

PARSEKIT_RESULT_VAR (entier)

Flag d'un noeud opcode
 PARSEKIT_USAGE_UNKNOWNN (entier)
 Flag d'un noeud opcode
 PARSEKIT_ZEND_INTERNAL_CLASS (entier)
 Type de la classe
 PARSEKIT_ZEND_USER_CLASS (entier)
 Type de la classe
 PARSEKIT_ZEND_EVAL_CODE (entier)
 Type de la fonction
 PARSEKIT_ZEND_INTERNAL_FUNCTION (entier)
 Type de la fonction
 PARSEKIT_ZEND_OVERLOADED_FUNCTION (entier)
 Type de la fonction
 PARSEKIT_ZEND_OVERLOADED_FUNCTION_TEMPORARY (entier) PHP >= 5.0.0
 Type de la fonction
 PARSEKIT_ZEND_USER_FUNCTION (entier)
 Type de la fonction
 PARSEKIT_IS_CONST (entier)
 Type du noeud
 PARSEKIT_IS_TMP_VAR (entier)
 Type du noeud
 PARSEKIT_IS_UNUSED (entier)
 Type du noeud
 PARSEKIT_IS_VAR (entier)
 Type du noeud
 PARSEKIT_ZEND_ADD (entier)
 Opcode
 PARSEKIT_ZEND_ADD_ARRAY_ELEMENT (entier)
 Opcode
 PARSEKIT_ZEND_ADD_CHAR (entier)
 Opcode
 PARSEKIT_ZEND_ADD_INTERFACE (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_ADD_STRING (entier)
 Opcode
 PARSEKIT_ZEND_ADD_VAR (entier)
 Opcode
 PARSEKIT_ZEND_ASSIGN (entier)
 Opcode
 PARSEKIT_ZEND_ASSIGN_ADD (entier)
 Opcode
 PARSEKIT_ZEND_ASSIGN_BW_AND (entier)
 Opcode
 PARSEKIT_ZEND_ASSIGN_BW_OR (entier)
 Opcode
 PARSEKIT_ZEND_ASSIGN_BW_XOR (entier)
 Opcode
 PARSEKIT_ZEND_ASSIGN_CONCAT (entier)
 Opcode
 PARSEKIT_ZEND_ASSIGN_DIM (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_ASSIGN_DIV (entier)
 Opcode
 PARSEKIT_ZEND_ASSIGN_MOD (entier)

Opcode
 PARSEKIT_ZEND_ASSIGN_MUL (entier)
 Opcode
 PARSEKIT_ZEND_ASSIGN_OBJ (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_ASSIGN_REF (entier)
 Opcode
 PARSEKIT_ZEND_ASSIGN_SL (entier)
 Opcode
 PARSEKIT_ZEND_ASSIGN_SR (entier)
 Opcode
 PARSEKIT_ZEND_ASSIGN_SUB (entier)
 Opcode
 PARSEKIT_ZEND_BEGIN_SILENCE (entier)
 Opcode
 PARSEKIT_ZEND_BOOL (entier)
 Opcode
 PARSEKIT_ZEND_BOOL_NOT (entier)
 Opcode
 PARSEKIT_ZEND_BOOL_XOR (entier)
 Opcode
 PARSEKIT_ZEND_BRK (entier)
 Opcode
 PARSEKIT_ZEND_BW_AND (entier)
 Opcode
 PARSEKIT_ZEND_BW_NOT (entier)
 Opcode
 PARSEKIT_ZEND_BW_OR (entier)
 Opcode
 PARSEKIT_ZEND_BW_XOR (entier)
 Opcode
 PARSEKIT_ZEND_CASE (entier)
 Opcode
 PARSEKIT_ZEND_CAST (entier)
 Opcode
 PARSEKIT_ZEND_CATCH (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_CLONE (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_CONCAT (entier)
 Opcode
 PARSEKIT_ZEND_CONT (entier)
 Opcode
 PARSEKIT_ZEND_DECLARE_CLASS (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_DECLARE_FUNCTION (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_DECLARE_INHERITED_CLASS (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_DIV (entier)
 Opcode
 PARSEKIT_ZEND_DO_FCALL (entier)
 Opcode
 PARSEKIT_ZEND_DO_FCALL_BY_NAME (entier)

Opcode
 PARSEKIT_ZEND_ECHO (entier)
 Opcode
 PARSEKIT_ZEND_END_SILENCE (entier)
 Opcode
 PARSEKIT_ZEND_EXIT (entier)
 Opcode
 PARSEKIT_ZEND_EXT_FCALL_BEGIN (entier)
 Opcode
 PARSEKIT_ZEND_EXT_FCALL_END (entier)
 Opcode
 PARSEKIT_ZEND_EXT_NOP (entier)
 Opcode
 PARSEKIT_ZEND_EXT_STMT (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_CLASS (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_FETCH_CONSTANT (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_DIM_FUNC_ARG (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_DIM_IS (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_DIM_R (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_DIM_RW (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_DIM_TMP_VAR (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_DIM_UNSET (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_DIM_W (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_FUNC_ARG (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_IS (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_OBJ_FUNC_ARG (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_OBJ_IS (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_OBJ_R (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_OBJ_RW (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_OBJ_UNSET (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_OBJ_W (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_R (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_RW (entier)
 Opcode
 PARSEKIT_ZEND_FETCH_UNSET (entier)

Opcode
 PARSEKIT_ZEND_FETCH_W (entier)
 Opcode
 PARSEKIT_ZEND_FE_FETCH (entier)
 Opcode
 PARSEKIT_ZEND_FE_RESET (entier)
 Opcode
 PARSEKIT_ZEND_FREE (entier)
 Opcode
 PARSEKIT_ZEND_HANDLE_EXCEPTION (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_IMPORT_CLASS (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_IMPORT_CONST (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_IMPORT_FUNCTION (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_INCLUDE_OR_EVAL (entier)
 Opcode
 PARSEKIT_ZEND_INIT_ARRAY (entier)
 Opcode
 PARSEKIT_ZEND_INIT_CTOR_CALL (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_INIT_FCALL_BY_NAME (entier)
 Opcode
 PARSEKIT_ZEND_INIT_METHOD_CALL (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_INIT_STATIC_METHOD_CALL (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_INIT_STRING (entier)
 Opcode
 PARSEKIT_ZEND_INSTANCEOF (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_ISSET_ISEMPTY (entier) PHP < 5.0.0
 Opcode
 PARSEKIT_ZEND_ISSET_ISEMPTY_DIM_OBJ (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_ISSET_ISEMPTY_PROP_OBJ (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_ISSET_ISEMPTY_VAR (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_IS_EQUAL (entier)
 Opcode
 PARSEKIT_ZEND_IS_IDENTICAL (entier)
 Opcode
 PARSEKIT_ZEND_IS_NOT_EQUAL (entier)
 Opcode
 PARSEKIT_ZEND_IS_NOT_IDENTICAL (entier)
 Opcode
 PARSEKIT_ZEND_IS_SMALLER (entier)
 Opcode
 PARSEKIT_ZEND_IS_SMALLER_OR_EQUAL (entier)
 Opcode
 PARSEKIT_ZEND_JMP (entier)

Opcode
 PARSEKIT_ZEND_JMPNZ (entier)
 Opcode
 PARSEKIT_ZEND_JMPNZ_EX (entier)
 Opcode
 PARSEKIT_ZEND_JMPZ (entier)
 Opcode
 PARSEKIT_ZEND_JMPZNX (entier)
 Opcode
 PARSEKIT_ZEND_JMPZ_EX (entier)
 Opcode
 PARSEKIT_ZEND_JMP_NO_CTOR (entier)
 Opcode
 PARSEKIT_ZEND_MOD (entier)
 Opcode
 PARSEKIT_ZEND_MUL (entier)
 Opcode
 PARSEKIT_ZEND_NEW (entier)
 Opcode
 PARSEKIT_ZEND_NOP (entier)
 Opcode
 PARSEKIT_ZEND_OP_DATA (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_POST_DEC (entier)
 Opcode
 PARSEKIT_ZEND_POST_DEC_OBJ (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_POST_INC (entier)
 Opcode
 PARSEKIT_ZEND_POST_INC_OBJ (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_PRE_DEC (entier)
 Opcode
 PARSEKIT_ZEND_PRE_DEC_OBJ (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_PRE_INC (entier)
 Opcode
 PARSEKIT_ZEND_PRE_INC_OBJ (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_PRINT (entier)
 Opcode
 PARSEKIT_ZEND_QM_ASSIGN (entier)
 Opcode
 PARSEKIT_ZEND_RAISE_ABSTRACT_ERROR (entier) PHP >= 5.0.0
 Opcode
 PARSEKIT_ZEND_RECV (entier)
 Opcode
 PARSEKIT_ZEND_RECV_INIT (entier)
 Opcode
 PARSEKIT_ZEND_RETURN (entier)
 Opcode
 PARSEKIT_ZEND_SEND_REF (entier)
 Opcode
 PARSEKIT_ZEND_SEND_VAL (entier)

Opcode
PARSEKIT_ZEND_SEND_VAR (entier)
Opcode
PARSEKIT_ZEND_SEND_VAR_NO_REF (entier)
Opcode
PARSEKIT_ZEND_SL (entier)
Opcode
PARSEKIT_ZEND_SR (entier)
Opcode
PARSEKIT_ZEND_SUB (entier)
Opcode
PARSEKIT_ZEND_SWITCH_FREE (entier)
Opcode
PARSEKIT_ZEND_THROW (entier) PHP >= 5.0.0
Opcode
PARSEKIT_ZEND_TICKS (entier)
Opcode
PARSEKIT_ZEND_UNSET_DIM_OBJ (entier)
Opcode
PARSEKIT_ZEND_UNSET_VAR (entier)
Opcode
PARSEKIT_ZEND_VERIFY_ABSTRACT_CLASS (entier) PHP >= 5.0.0
Opcode

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [parsekit_compile_file](#)
- [parsekit_compile_string](#)
- [parsekit_func_arginfo](#)

8.97.7 parsekit_compile_string() : Compile une chaîne de code PHP et retourne le tableau d'opcode résultant

array **parsekit_compile_string** (string *phpcode* , array *errors* , int *options*)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

8.97.8 parsekit_func_arginfo() : Retourne des informations concernant les arguments d'une fonction

array **parsekit_func_arginfo** (mixed *function*)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

8.98 Contrôle des processus

8.98.1 Introduction

Le système de contrôle des processus de PHP implémente un système de création, gestion et terminaison des processus comme sous Unix. Cette extension ne doit pas être activée pour une utilisation en serveur web, car les résultats pourraient être inattendus.

Cette documentation explique l'utilisation générale des fonctions du gestionnaire de processus. Pour des informations plus détaillées sur le contrôle des processus Unix, nous vous encourageons à consulter votre documentation système, incluant notamment `fork(2)`, `waitpid(2)` et `signal(2)`, ou bien à consulter un ouvrage de référence comme "Advanced Programming in the UNIX Environment" par W. Richard Stevens (Addison-Wesley).

PCNTL utilise maintenant les ticks comme mécanisme de callback du gestionnaire de signaux, ce qui est beaucoup plus rapide que la version précédente. Ce changement suit la même sémantique que l'utilisation de "ticks utilisateurs". Vous utilisez `declare` pour spécifier les endroits de votre programme où les callbacks peuvent être appelés. Cela vous permet de minimaliser la consommation due à la gestion d'événements asynchrones. Par le passé, compiler PHP avec `pcntl` revenait toujours à souffrir cette consommation, même si votre script n'utilisait pas `pcntl`.

Il y'a un ajustement à faire pour tous les scripts `pcntl` basés sur une version antérieure à PHP 4.3.0 afin qu'ils puissent fonctionner : il faut soit utiliser `declare` dans la section où vous voulez avoir le callback, soit l'activer pour tout le script avec la nouvelle syntaxe globale de `declare` .

Note

Cette extension n'est pas disponible sur les plates-formes Windows.

8.98.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.98.3 Installation

Le support du contrôle des processus n'est pas activé par défaut en PHP. Vous devez compiler la version CLI ou CGI de PHP avec l'option de configuration `--enable-pcntl` pour activer le support de cette extension.

Note

Actuellement, ce module ne fonctionne pas sur les environnements non Unix comme Windows.
--

8.98.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.98.5 Types de ressources

Cette extension ne définit aucune ressource.

8.98.6 Constantes pré-définies

La liste suivante représente les signaux supportés par les fonctions de gestion des processus. Reportez-vous au manuel de votre système (signal(7)) pour plus de détails sur ces signaux.

WNOHANG ([entier](#))
WUNTRACED ([entier](#))
SIG_IGN ([entier](#))
SIG_DFL ([entier](#))
SIG_ERR ([entier](#))
SIGHUP ([entier](#))
SIGINT ([entier](#))
SIGQUIT ([entier](#))
SIGILL ([entier](#))
SIGTRAP ([entier](#))
SIGABRT ([entier](#))
SIGIOT ([entier](#))
SIGBUS ([entier](#))
SIGFPE ([entier](#))
SIGKILL ([entier](#))
SIGUSR1 ([entier](#))
SIGSEGV ([entier](#))
SIGUSR2 ([entier](#))
SIGPIPE ([entier](#))
SIGALRM ([entier](#))
SIGTERM ([entier](#))
SIGSTKFLT ([entier](#))
SIGCLD ([entier](#))
SIGCHLD ([entier](#))
SIGCONT ([entier](#))
SIGSTOP ([entier](#))
SIGTSTP ([entier](#))
SIGTTIN ([entier](#))
SIGTTOU ([entier](#))
SIGURG ([entier](#))
SIGXCPU ([entier](#))
SIGXFSZ ([entier](#))
SIGVTALRM ([entier](#))
SIGPROF ([entier](#))
SIGWINCH ([entier](#))
SIGPOLL ([entier](#))
SIGIO ([entier](#))
SIGPWR ([entier](#))
SIGSYS ([entier](#))
SIGBABY ([entier](#))

8.98.7 Exemples

Cet exemple forke un processus démon, avec un gestionnaire de signaux.

Exemple de contrôle de processus

[<?php](#)

```

$pid = pcntl_fork();
if ($pid == -1) {
    die("impossible de forker");
} elseif ($pid) {
    exit(); // nous sommes le processus père
} else {
    //nous sommes le processus fils
}

// détachons le processus du terminal
if (!posix_setsid()) {
    die("impossible de se détacher du terminal");
}

// boucle infinie
while(1) {
    // exécution de quelque chose
}

function sig_handler($signo)
{
    switch($signo) {
        case SIGTERM:
            // gestion des tâches de terminaison
            exit;
            break;
        case SIGHUP:
            // gestion des tâches de redémarrage
            break;
        default:
            // gestion des autres tâches
    }
}

// configuration des gestinnaires de signaux
pcntl_signal(SIGTERM, "sig_handler");
pcntl_signal(SIGHUP, "sig_handler");

?>

```

8.98.8 Voir aussi

Un tour par la section sur les fonctions [POSIX](#) peut être utile.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [Voir aussi](#)
- [pcntl_alarm](#)
- [pcntl_exec](#)
- [pcntl_fork](#)
- [pcntl_getpriority](#)
- [pcntl_setpriority](#)
- [pcntl_signal](#)
- [pcntl_wait](#)

- [pcntl_waitpid](#)
- [pcntl_wexitstatus](#)
- [pcntl_wifexited](#)
- [pcntl_wifsignaled](#)
- [pcntl_wifstopped](#)
- [pcntl_wstopsig](#)
- [pcntl_wtermsig](#)

8.98.10 [pcntl_exec\(\)](#) : Exécute le programme indiqué dans l'espace courant de processus

```
void pcntl\_exec ( string path , array args , array envs )
```

[pcntl_exec](#) exécute le programme path avec les arguments args . path doit être le chemin vers un binaire exécutable ou un script avec un chemin valide pointant vers un exécutable à la première ligne (par exemple `#!/usr/local/bin/perl`). Voir les pages d'aide de votre système concernant `execve(2)` pour plus d'informations.

args est un tableau d'arguments sous la forme de chaînes de caractères passées au programme.

envs est un tableau de chaînes de caractères qui sont passées au programme comme variables d'environnement. Le tableau est de la forme nom => valeur, la clé est le nom de la variable d'environnement et la valeur est la valeur de cette variable.

[pcntl_exec](#) retourne FALSE en cas d'erreur et ne retourne rien en cas de succès.

8.98.11 [pcntl_fork\(\)](#) : Duplique le process courant

```
int pcntl\_fork ( void )
```

[pcntl_fork](#) crée un processus fils, qui ne diffère du processus père que par l'identifiant de processus et l'identifiant PPID. Reportez-vous à la page de man `fork(2)` pour avoir des détails sur le comportement de cette fonction sur votre système.

En cas de succès, le PID (identifiant de processus) du fils est retourné dans le processus père, et 0 est retourné dans le processus fils. En cas d'échec, -1 est retourné dans le contexte du père, aucun processus fils ne sera créé et PHP générera une erreur.

Exemple avec [pcntl_fork](#)

```
<?php
$pid = pcntl\_fork();
if ($pid == -1) {
    die('dupplication impossible');
} else if ($pid) {
    // le père
    pcntl\_wait($status); //Protège encore des enfants zombies
} else {
    // le fils
}
?>
```

Voir aussi [pcntl_waitpid](#) et [pcntl_signal](#) .

8.98.12 `pcntl_getpriority()` : Retourne la priorité d'un processus

`int pcntl_getpriority (int pid , int process_identifieur)`

`pcntl_getpriority` retourne la priorité de `pid` . Si `pid` n'est pas spécifié, le `pid` du processus courant est utilisé. Comme les niveaux de priorités changent entre les types de systèmes et les versions de kernel, lisez la page de manuel `getpriority(2)` de votre système pour des détails spécifiques.

`pcntl_getpriority` retourne la priorité du processus ou `FALSE` en cas d'erreur. Une valeur numérique plus petite rend la planification plus favorable.

Le paramètre `process_identifieur` peut prendre comme valeur `PRIO_PGRP` , `PRIO_USER` ou `PRIO_PROCESS` .

Attention

Cette fonction peut retourner `FALSE` , mais elle peut aussi retourner une valeur équivalent à `FALSE` , utilisable dans une condition `if` simple. Utilisez l' opérateur `===` pour tester la valeur de retour exacte de cette fonction.

8.98.13 `pcntl_setpriority()` : Change la priorité d'un processus

`bool pcntl_setpriority (int priority , int pid , int process_identifieur)`

`pcntl_setpriority` change la priorité de `pid` à `priority` . Si `pid` n'est pas spécifié, le `pid` du processus courant est utilisé.

`priority` est généralement une valeur allant de -20 à 20 . La priorité par défaut est 0 tandis qu'une valeur numérique plus petite favorise une meilleure planification. Comme les niveaux de priorité changent entre les types de systèmes et les versions de kernel, lisez la page de manuel `getpriority(2)` de votre système pour des détails spécifiques.

Le paramètre `process_identifieur` peut prendre comme valeur `PRIO_PGRP` , `PRIO_USER` or `PRIO_PROCESS` .

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

8.98.14 `pcntl_signal()` : Installe un gestionnaire de signaux

`bool pcntl_signal (int signo , callback handle , bool restart_syscalls)`

`pcntl_signal` installe un nouveau gestionnaire de signaux pour le signal indiqué par le paramètre `signo` . Le gestionnaire de signaux est affecté à `handler` qui peut être le nom d'une fonction utilisateur, ou bien l'une des deux constantes globales `SIG_IGN` et `SIG_DFL` . Le paramètre optionnel `restart_syscalls` spécifie si l'appel système de redémarrage (`restarting`) doit être utilisé lorsque ce signal arrive et, par défaut, ce paramètre vaut `TRUE` .

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Note

Le paramètre optionnel `restart_syscalls` n'est disponible que depuis la version 4.3.0 de PHP.

Note

La possibilité d'utiliser une méthode d'objet comme fonction de callback est disponible depuis PHP 4.3.0. Notez que lorsque vous configurez le gestionnaire avec une méthode d'objet, le compteur de référence de l'objet est incrémenté, ce qui le rend persistant jusqu'à ce que vous changiez le gestionnaire de signaux pour un autre, ou que le script se termine.

Exemple avec `pcntl_signal`

```
<?php
// l'usage des ticks est nécessaire depuis PHP 4.3.0
declare (ticks = 1);

// gestionnaire de signaux système
function sig_handler($signo)
{
    switch($signo) {
        case SIGTERM:
            // gestion de l'extinction
            exit;
            break;
        case SIGHUP:
            // gestion du redémarrage
            break;
        case SIGUSR1:
            echo "Reçu le signe SIGUSR1...\n";
            break;
        default:
            // gestion des autres signaux
    }
}

echo "Installation du gestionnaire de signaux...\n";

// Installation des gestionnaires de signaux
pcntl_signal(SIGTERM, "sig_handler");
pcntl_signal(SIGHUP, "sig_handler");
pcntl_signal(SIGUSR1, "sig_handler");

// ou bien utilisez un objet (disponible depuis PHP 4.3.0)
// pcntl_signal(SIGUSR1, array($obj, "faire_quelque_chose"));

echo "Génération d'un signal SIGTERM à moi-même...\n";

// envoi de SIGUSR1 à l'identifiant de processus courant
posix_kill(posix_getpid(), SIGUSR1);

echo "Done\n"

?>
```

Note

Depuis PHP 4.3.0, PCNTL utilise les ticks comme mécanisme de signaux de traitement des rappels qui est plus rapide que l'ancien mécanisme. Ce changement suit les mêmes règles que l'utilisation des " `user ticks` ". Vous devriez utiliser la requête `declare()` pour spécifier l'endroit dans votre programme où les rappels sont autorisés à être utilisés pour le traitement du signal de la fonction proprement dite (comme utilisé dans l'exemple précédent).

Voir aussi `pcntl_fork` et `pcntl_waitpid` .

8.98.15 `pcntl_wait()` : Attend ou retourne le statut d'un processus fils

`int pcntl_wait (int status , int options)`

`pcntl_wait` suspend l'exécution du processus courant jusqu'à ce qu'un des processus fils soit terminé, ou qu'un signal soit envoyé pour terminer le processus courant ou pour appeler un gestionnaire. Si le processus est déjà terminé au moment de l'appel de la fonction, c'est-à-dire si le processus est un zombie, alors la fonction se termine immédiatement. Toutes les ressources système utilisées par le processus fils sont libérées. Lisez le manuel de votre système à `wait(2)` pour avoir des détails spécifiques sur le fonctionnement de `wait()` sur celui-ci.

`pcntl_wait` retourne l'identifiant de processus qui s'est terminé, -1 en cas d'erreur ou zéro si `WNOHANG` a été fourni comme option (disponible sur les systèmes `wait3`), et qu'aucun processus fils n'était disponible.

Si `wait3` est disponible sur votre système (c'est le cas de la plupart des systèmes BSD-), vous pouvez ajouter le paramètre optionnel `options`. S'il n'est pas fourni, `wait()` sera utilisé pour l'appel système. Si `wait3` n'est pas disponible, le paramètre `options` n'aura pas d'effet. La valeur de `options` est la combinaison de zéro ou plusieurs des constantes suivantes :

<code>WNOHANG</code>	Termine immédiatement si aucun processus ne s'est terminé.
<code>WUNTRACED</code>	Termine pour les processus qui sont stoppés, et pour ceux dont le résultat n'a pas été rapporté.

`pcntl_wait` va stocker les informations de statut dans le paramètre `status` qui peut être lu avec les fonctions suivantes : `pcntl_wifexited`, `pcntl_wifstopped`, `pcntl_wifsignaled`, `pcntl_wexitstatus`, `pcntl_wtermsig` et `pcntl_wstopsig`.

Note
Cette fonction est l'équivalent de l'appel de <code>pcntl_waitpid</code> avec la valeur -1 <code>pid</code> et sans paramètre <code>options</code> .

Voir aussi `pcntl_fork`, `pcntl_signal`, `pcntl_wifexited`, `pcntl_wifstopped`, `pcntl_wifsignaled`, `pcntl_wexitstatus`, `pcntl_wtermsig`, `pcntl_wstopsig` et `pcntl_waitpid`.

8.98.16 `pcntl_waitpid()` : Attend la fin de l'exécution d'un processus fils

`int pcntl_waitpid (int pid , int status , int options)`

`pcntl_waitpid` suspend l'exécution du processus courant jusqu'à ce qu'un processus fils spécifié par le paramètre `pid` ait terminé, qu'un signal ait mis fin à ce processus ou qu'un signal ait appelé un gestionnaire de signaux. Si le processus fils identifié par `pid` est déjà terminé au moment de l'appel de cette fonction (on les appelle des processus "zombie"), la fonction se termine immédiatement. Toute ressource système utilisée par le processus fils est libérée. Reportez-vous à la page de man `waitpid(2)` pour avoir des détails sur le comportement de cette fonction sur votre système.

`pcntl_waitpid` retourne l'identifiant de processus du processus fils qui s'est terminé, ou bien -1 en cas d'erreur ou encore zéro si `WNOHANG` a été utilisée et qu'aucun processus fils n'était disponible.

Le paramètre `pid` peut prendre l'une des valeurs suivantes :

< -1	attend que tous les processus fils dont l'identifiant de groupe est égal à la valeur absolue de <code>pid</code> soient terminés.
------	---

-1	attend que tous les processus fils soient terminés. Ceci est le même comportement que celui de la fonction <u>wait</u> .
0	attend que tous les processus fils dont l'identifiant de groupe est égal à celui du processus courant soient terminés.
> 0	attend que le processus fils dont l'identifiant est égal à pid soit terminé.

Note

Si pid vaut -1 , cela équivaut à utiliser la fonction pcntl_wait (minimum options) .

pcntl_waitpid enregistrera des informations sur le statut courant du processus dans le paramètre status , auquel on peut accéder grace aux fonctions suivantes : pcntl_wifexited , pcntl_wifstopped , pcntl_wifsignaled , pcntl_wexitstatus , pcntl_wtermsig et pcntl_wstopsig .

Le paramètre options peut prendre la valeur zéro, ou plusieurs des constantes globales suivantes (combinez les avec l'opérateur OR) :

WNOHANG	retourne immédiatement si aucun processus fils ne s'est terminé.
WUNTRACED	retourne lorsque les processus fils sont arrêtés et que leur statut n'a pas été mis à jour.

Voir aussi pcntl_fork , pcntl_signal , pcntl_wifexited , pcntl_wifstopped , pcntl_wifsignaled , pcntl_wexitstatus , pcntl_wtermsig et pcntl_wstopsig .

8.98.17 pcntl_wexitstatus() : Retourne le code d'un processus fils terminé

int pcntl_wexitstatus (int status)

pcntl_wexitstatus retourne le code de retour du processus fils. Cette fonction n'est utile que si la fonction pcntl_wifexited a retourné TRUE .

Le paramètre status est le paramètre status passé à un appel de pcntl_waitpid ayant réussi.

Voir aussi pcntl_waitpid et pcntl_wifexited .

8.98.18 pcntl_wifexited() : Retourne TRUE si le code de retour représente une fin normale

bool pcntl_wifexited (int status)

pcntl_wifexited retourne TRUE si le processus fils a retourné un code qui représente une fin normale.

Le paramètre status est le paramètre status passé à un appel de pcntl_waitpid ayant réussi.

Voir aussi pcntl_waitpid et pcntl_wexitstatus .

8.98.19 pcntl_wifsignaled() : Retourne TRUE si le code statut représente une fin due à un signal

bool pcntl_wifsignaled (int status)

pcntl_wifsignaled retourne TRUE si le processus fils sort car un signal n'a pu être reçu.

Le paramètre status est le paramètre status passé à un appel de pcntl_waitpid ayant réussi.

Voir aussi pcntl_waitpid et pcntl_signal .

8.98.20 pcntl_wifstopped() : Retourne TRUE si le processus fils est arrêté

bool **pcntl_wifstopped** (int status)

pcntl_wifstopped retourne TRUE si le processus fils est arrêté ; cela n'est possible que si l'appel à pcntl_waitpid a été fait avec l'option WUNTRACED .

Le paramètre status est le paramètre status passé à un appel de pcntl_waitpid ayant réussi.

Voir aussi pcntl_waitpid .

8.98.21 pcntl_wstopsig() : Retourne le signal qui a causé l'arrêt du processus fils

int **pcntl_wstopsig** (int status)

pcntl_wstopsig retourne le numéro du signal qui a causé l'arrêt du processus fils. Cette fonction est uniquement utile si pcntl_wifstopped a retourné TRUE .

Le paramètre status est le paramètre status passé à un appel de pcntl_waitpid ayant réussi.

Voir aussi pcntl_waitpid et pcntl_wifstopped .

8.98.22 pcntl_wtermsig() : Retourne le signal qui a provoqué la fin du processus fils

int **pcntl_wtermsig** (int status)

pcntl_wtermsig retourne le numéro du signal qui a provoqué la fin du processus fils. Cette fonction est uniquement utile si pcntl_wifsignaled retourne TRUE .

Le paramètre status est le paramètre status passé à un appel de pcntl_waitpid ayant réussi.

Voir aussi pcntl_waitpid , pcntl_signal et pcntl_wifsignaled .

8.99 Expressions rationnelles compatibles Perl

8.99.1 Introduction

La syntaxe des masques utilisés dans ces fonctions ressemble fort à celle de Perl. Les expressions seront entourées de délimiteurs, slash (/), par exemple. N'importe quel caractère peut servir de délimiteur, tant qu'il n'est pas alpha-numérique ou n'est pas un anti-slash (\). Si un délimiteur doit être utilisé dans l'expression, il faudra l'échapper avec un anti-slash. Depuis PHP 4.0.4, vous pouvez utiliser les délimiteurs (), {}, [], et <>, comme en Perl. Voir la [syntaxe des masques](#) pour plus d'explications.

Le délimiteur final peut être suivi d'options qui affecteront la recherche. Voir aussi [options de recherche](#) .

PHP supporte également les expressions rationnelles utilisant la syntaxe POSIX étendue ([fonctions REGEX POSIX-extended](#)).

Attention

Vous devez être conscient des limitations de PCRE. Lisez http://www.pcre.org/pcre.txt pour plus de détails.
--

8.99.2 Pré-requis

Le support des expressions rationnelles est assuré par la bibliothèque PCRE, qui est open source, et écrite par Philip Hazel. Elle est soumise au copyright de l'University of Cambridge, Angleterre. Elle est disponible à <ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/> .

8.99.3 Installation

A partir de PHP 4.2.0, ces fonctions sont activées par défaut. Pour les anciennes versions, vous devez configurer et compiler PHP avec l'option `--with-pcre-regex` afin de pouvoir les utiliser. Utilisez `--with-pcre-regex=DIR` pour spécifier le répertoire où la bibliothèque PCRE est installée si vous n'utilisez pas la bibliothèque incluse dans PHP. Pour les versions plus anciennes de PHP, vous devez configurer et compiler PHP avec `--with-pcre-regex[=DIR]` pour pouvoir utiliser ces fonctions.

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de bibliothèque supplémentaire pour disposer de ces fonctions.

8.99.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.99.5 Types de ressources

Cette extension ne définit aucune ressource.

8.99.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Constante	Description
PREG_PATTERN_ORDER	Ordonne les résultats de façon à ce que <code>\$matches[0]</code> contienne les résultats qui correspondent au masque entier, <code>\$matches[1]</code> ceux qui correspondent à la première parenthèse capturante, <code>\$matches[2]</code> ceux qui correspondent à la deuxième parenthèse capturante, etc. Cette constante est utilisée avec <code>preg_match_all</code> .
PREG_SET_ORDER	Les résultats sont classés de telle façon que <code>\$matches[0]</code> contient la première série de résultat, <code>\$matches[1]</code> la deuxième, etc. Cette constante est utilisée avec <code>preg_match_all</code> .
PREG_OFFSET_CAPTURE	Voir la description de <code>PREG_SPLIT_OFFSET_CAPTURE</code> . Cette constante est utilisée depuis PHP 4.3.0.
PREG_SPLIT_NO_EMPTY	Si cette option est activée, seules les sous-chaînes non vides seront retournées par <code>preg_split</code> .
PREG_SPLIT_DELIM_CAPTURE	Si cette option est activée, les expressions entre parenthèses entre les délimiteurs de masques seront aussi capturées et retournées. Cette option a été ajoutée en PHP 4.0.5. Cette constante est utilisée avec <code>preg_split</code> .
PREG_SPLIT_OFFSET_CAPTURE	Si cette constante est utilisée avec <code>preg_split</code> , l'offset de début de résultat sera retourné, en plus de la chaîne résultat. Notez que cela change la nature du résultat retourné en un tableau contenant une chaîne à l'offset 0 et une chaîne contenant un offset à l'offset 1. Cette option est disponible depuis PHP 4.3.0.

8.99.7 Exemples

Exemples de masques valides
<ul style="list-style-type: none"> • <code></\w+>/</code> • <code> (\d{3})-\d+ Sm</code> • <code>/^(?i)php[34]/</code> • <code>{^\s+(\s+)?\$}</code>

Exemples de masques invalides
<ul style="list-style-type: none"> • <code>/href='(.*)'</code> - délimiteur final manquant • <code>/w+\s*\w+/J</code> - option 'J' inconnue • <code>1-\d3-\d3-\d4 </code> - délimiteur initial manquant

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)

- [Exemples](#)
- [options de recherche](#)
- [syntaxe des masques](#)
- [preg_grep](#)
- [preg_match_all](#)
- [preg_match](#)
- [preg_quote](#)
- [preg_replace_callback](#)
- [preg_replace](#)
- [preg_split](#)

8.99.9 [syntaxe des masques\(\)](#) : Fonctionnement des expressions rationnelles

La bibliothèque PCRE est un ensemble de fonctions qui implémentent la recherche par expressions rationnelles, en utilisant la même syntaxe et la même sémantique que le Perl 5, avec quelques nuances (voir ci-dessous). L'implémentation actuelle est celle de Perl 5.005.

8.99.10 [preg_grep\(\)](#) : Retourne un tableau avec les résultats de la recherche

array [preg_grep](#) (string pattern , array input , int flags)

[preg_grep](#) retourne un tableau qui contient les éléments de input qui satisfont le masque pattern .

flags peut être :

`PREG_GREP_INVERT`

Si cette option est passée, [preg_grep](#) retourne les éléments du tableau input qui ne correspondent **pas** au masque pattern . Cette option est disponible depuis PHP 4.2.0.

Depuis PHP 4.0.4, le tableau retourné par [preg_grep](#) est indexé en utilisant les clés issues du tableau input . Si ces clés sont inutiles, utilisez la fonction [array_values](#) sur le tableau retourné par [preg_grep](#) pour obtenir le comportement traditionnel.

Exemple avec [preg_grep](#)

```
<?php
// recherche les nombres à virgule flottante dans le tableau
preg_grep("/^(\d+)?\.\d+$/", $array);
?>
```

8.99.11 [preg_match_all\(\)](#) : Expression rationnelle globale

int [preg_match_all](#) (string pattern , string subject , array matches , int flags , int offset)

[preg_match_all](#) analyse subject pour trouver l'expression pattern et met les résultats dans matches , dans l'ordre spécifié par flags .

Après avoir trouvé un premier résultat, la recherche continue jusqu'à la fin de la chaîne.

flags peut prendre une des deux valeurs suivantes (notez bien qu'il est incohérent d'utiliser PREG_PATTERN_ORDER avec PREG_SET_ORDER) :

PREG_PATTERN_ORDER

L'ordre est tel que \$matches[0] est un tableau qui contient les résultats qui satisfont le masque complet, \$matches[1] est un tableau qui contient les résultats qui satisfont la première parenthèse capturante, etc.

Exemple avec preg_match_all et PREG_PATTERN_ORDER

```
<?php
preg_match_all("|<[^>]+>(.*?)</[^>]+>|U",
    "<strong>exemple : </strong><div align=\"left\">un test</div>",
    $out,
    PREG_PATTERN_ORDER);

echo $out[0][0] . ', ' . $out[0][1] . "\n";
echo $out[1][0] . ', ' . $out[1][1] . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
<strong>exemple : </strong>, <div align="left">un test</div>
exemple : , un test
```

Ainsi, \$out[0] est un tableau qui contient les résultats qui satisfont le masque complet, et \$out[1] est un tableau qui contient les balises entre > et <.

PREG_SET_ORDER

Les résultats sont classés de telle façon que \$matches[0] contient la première série de résultats, \$matches[1] contient la deuxième, etc.

Exemple avec preg_match_all et PREG_SET_ORDER

```
<?php
preg_match_all("|<[^>]+>(.*?)</[^>]+>|U",
    "<strong>exemple : </strong><div align=\"left\">un test</div>",
    $out,
    PREG_SET_ORDER);

echo $out[0][0] . ', ' . $out[0][1] . "\n";
echo $out[1][0] . ', ' . $out[1][1] . "\n";
?>
```

L'exemple ci-dessus va afficher :

```
<strong>exemple : </strong>, exemple :
<div align="left">un test</div>, un test
```

Dans ce cas, \$matches[0] est la première série de résultat, et \$matches[0][0] contient le texte qui satisfait le masque complet, \$matches[0][1] contient le texte de la première parenthèse capturante, etc. De même, \$matches[1] contient le texte qui satisfait le masque complet, etc.

PREG_OFFSET_CAPTURE

Si cette option est activée, toutes les sous-chaînes qui satisfont le masque seront aussi identifiées par leur offset. Notez que cela modifie le format de la valeur retournée, puisque chaque élément de réponse devient un tableau contenant la sous-chaîne résultat, à l'index 0 dans la chaîne subject constant 1. Cette option a été ajoutée en PHP 4.3.0.

Si order est omis, PREG_PATTERN_ORDER est utilisé par défaut.

Normalement, la recherche commence au début de la chaîne subject . Le paramètre optionnel offset peut être utilisé pour spécifier une position pour le début de la recherche. Le paramètre offset a été introduit depuis la version 4.3.3 de PHP.

Note

Utiliser le paramètre offset ne revient pas à passer substr(\$subject, \$offset) à preg_match_all à la place de la chaîne subject , car pattern peut contenir des assertions comme ^ , \$ ou (?<=x) . Lisez la documentation sur la fonction preg_match pour des exemples.

preg_match_all retourne le nombre de résultats qui satisfont le masque complet, ou FALSE en cas d'échec ou d'erreur.

Extraction de tous les numéros de téléphone d'un texte

```
<?php
preg_match_all("/\(? (\d{3})? \)? (? (1) [\-\s] ) \d{3}-\d{4}/x",
                "Appelez 555-1212 ou 1-800-555-1212", $phones);
?>
```

Recherche les couples de balises HTML (gourmand)

```
<?php
// Cet exemple utilise les références arrières (\\2).
// Elles indiquent à l'analyseur qu'il doit trouver quelque chose qu'il
// a déjà repéré un peu plus tôt
// le nombre 2 indique que c'est le deuxième jeu de parenthèses
// capturant qui doit être utilisé (ici, ([\w]+)).
// L'antislash est nécessaire ici, car la chaîne est entre guillemets doubles

$html = '<strong>Texte en gras</strong><a href="salut.html">clique moi</a>';

preg_match_all("/(<([\w]+) [^>]*>)(.*)(<\/\\2>)/", $html, $matches, PREG_SET_ORDER);

foreach ($matches as $val) {
    echo "matched: " . $val[0] . "\n";
    echo "part 1: " . $val[1] . "\n";
    echo "part 2: " . $val[3] . "\n";
    echo "part 3: " . $val[4] . "\n\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
Trouvé : <strong>Texte en gras</strong>
partie 1 : <strong>
partie 2 : Texte en gras
partie 3 : </strong>
```

```
Trouvé : <a href="salut.html">clique moi</a>
partie 1 : <a href="salut.html">
partie 2 : clique moi
partie 3 : </a>
```

Voir aussi preg_match , preg_replace et preg_split .

8.99.12 preg_match() : Expression rationnelle standard

int **preg_match** (string pattern , string subject , array matches , int flags , int offset)

preg_match analyse subject pour trouver l'expression qui correspond à pattern .

Si matches est fourni, il sera rempli par les résultats de la recherche. \$matches[0] contiendra le texte qui satisfait le masque complet, \$matches[1] contiendra le texte qui satisfait la première parenthèse capturante, etc.

Le paramètre flags peut prendre l'une des valeurs suivantes :

PREG_OFFSET_CAPTURE

Si cette option est activée, toutes les sous-chaînes qui satisfont le masque seront aussi identifiées par leur offset. Notez que cela modifie le format de la valeur retournée , puisque chaque élément de réponse devient un tableau contenant la sous-chaîne résultat, à l'index 0 dans la chaîne subject constant 1 . Cette option a été ajoutée en PHP 4.3.0 .

Le paramètre flags a été ajouté en PHP 4.3.0 .

Normalement, la recherche commence au début de la chaîne subject . Le paramètre optionnel offset peut être utilisé pour spécifier une position pour le début de la recherche. Le paramètre offset a été introduit depuis la version 4.3.3 de PHP.

Note

Utiliser le paramètre offset ne revient pas à passer substr(\$subject, \$offset) à preg_match_all à la place de la chaîne subject , car pattern peut contenir des assertions comme ^ , \$ ou (?<=x) . Lisez la documentation sur la fonction preg_match pour des exemples. Comparez :

```
<?php
$subject = "abcdef";
$pattern = '/^def/';
preg_match($pattern, $subject, $matches, PREG_OFFSET_CAPTURE, 3);
print_r($matches);
?>
```

L'exemple ci-dessus va afficher :

Array

```
(
)
```

avec cet exemple :

```
<?php
$subject = "abcdef";
$pattern = '/^def/';
preg_match($pattern, substr($subject,3), $matches, PREG_OFFSET_CAPTURE);
print_r($matches);
?>
```

produira :

```

Array
(
  [0] => Array
  (
    [0] => def
    [1] => 0
  )
)

```

`preg_match` retourne le nombre de fois où le masque pattern a été trouvé. Cela peut aller de 0 (pas de solution) à un 1 car `preg_match` s'arrête dès qu'elle a trouvé une première solution. `preg_match_all`, au contraire, va continuer jusqu'à la fin de la chaîne subject. `preg_match` retourne FALSE si une erreur survient.

Array

Trouve la chaîne "php"

```

<?php
// Le "i" après le délimiteur du pattern indique que la recherche ne sera pas sensible à la casse
if (preg_match("/php/i", "PHP est le meilleur langage de script du web.)) {
    echo 'Un résultat a été trouvé.';
} else {
    echo 'Aucun résultat n\'a été trouvé.';
}
?>

```

Trouve le mot "web"

```

<?php
/* \b, dans le masque, indique une limite de mot, de façon à ce que le mot
"web" uniquement soit repéré, et pas seulement des parties de mots comme
dans "webbing" ou "cobweb" */
if (preg_match("/\bweb\b/i", "PHP est le meilleur langage de script du web.)) {
    echo 'Un mot a été trouvé.';
} else {
    echo 'Un mot n\'a pas été trouvé.';
}

if (preg_match("/\bweb\b/i", "PHP est le meilleur langage de script pour les webagency.)) {
    echo 'Un mot a été trouvé.';
} else {
    echo 'Un mot n\'a pas été trouvé.';
}
?>

```

Lire un nom de domaine dans une URL

```

<?php
// repérer le nom de l'hôte dans l'URL
preg_match("/^(http:\/\//)?([^\//]+)/i",
    "http://www.php.net/index.html", $matches);
$host = $matches[2];

// repérer les deux derniers segments du nom de l'hôte
preg_match("/[^\.\//]+\.[^\.\//]+$/", $host, $matches);
echo "Le nom de domaine est : ".$matches[0]."\n";
?>

```

L'exemple ci-dessus va afficher :

Le nom de domaine est : php.net

Voir aussi [preg_match_all](#) , [preg_replace](#) et [preg_split](#) .

8.99.13 preg_quote() : Echappement des caractères spéciaux des expressions rationnelles

string **preg_quote** (string *str* , string *delimiter*)

[preg_quote](#) ajoute un anti-slash devant tous les caractères de la chaîne *str* . Cela est très utile si vous avez une chaîne qui va servir de masque, mais qui est générée durant l'exécution.

Si l'argument optionnel *delimiter* est fourni, il sera aussi échappé. Ceci est pratique pour échapper le délimiteur requis par les fonctions PCRE. Le slash / est le délimiteur le plus répandu.

Les caractères spéciaux qui seront échappés : . \ + * ? [^] \$ () { } = ! < > | :

Exemple avec [preg_quote](#)

```
<?php
$keywords = '$40 pour un g3/400';
$keywords = preg_quote($keywords, '/');
echo $keywords; // retourne \$40 pour un g3\400
?>
```

Mise en gras d'un mot dans un texte

```
<?php
// Dans cet exemple, preg_quote($word) sert à éviter que les astérisques
// prennent une valeur particulière dans l'expression rationnelle.

$textbody = 'Ce livre est *très* difficile à trouver.';
$word = "*très*";
$textbody = preg_replace ("/" . preg_quote($word) . "/",
                          "<strong>" . $word . "</strong>",
                          $textbody);
?>
```

Note

Cette fonction gère les chaînes binaires.

8.99.14 preg_replace_callback() : Rechercher/remplacer avec une expression rationnelle et fonction de callback

mixed **preg_replace_callback** (mixed *pattern* , callback *callback* , mixed *subject* , int *limit* , int *count*)

Le comportement de [preg_replace_callback](#) est presque identique à celui de [preg_replace](#) , hormis le fait qu'à la place du paramètre remplacement , il faut spécifier une fonction de callback *callback* qui sera appelée, avec les éléments trouvés en arguments. Cette fonction retourne alors la chaîne de remplacement.

Exemple avec [preg_replace_callback](#)

```
<?php
// Ce texte était vrai en 2002
```

```
// nous voulons le mettre a jour pour 2003
$text = "Le premier avril est le 01/04/2002\n";
$text.= "Le dernier noël était le 25/12/2001\n";

// Fonction de callback
function next_year($matches)
{
    // comme d'habitude : $matches[0] représente la valeur totale
    // $matches[1] représente la première parenthèse capturante
    return $matches[1].($matches[2]+1);
}

echo preg_replace_callback(
    "|(\d{2}/\d{2}/)(\d{4})|",
    "next_year",
    $text);

// retourne ceci :
// Le premier avril est le 01/04/2003
// Le dernier noël était le 25/12/2002
?>
```

Vous aurez souvent besoin de la fonction callback avec `preg_replace_callback` à un seul endroit. Dans ce cas, vous pouvez simplement utiliser la fonction `create_function` pour déclarer une fonction anonyme comme fonction de callback pour `preg_replace_callback`. En faisant cela, vous concentrez toutes les routines liées à votre remplacement à un seul endroit, et nous ne polluez par votre espace de noms de fonctions avec des fonctions à usage unique.

Exemples avec `preg_replace_callback` et `create_function`

```
<?php
// Un filtre de ligne de commande Unix pour convertir les lettres
// en majuscule de début des paragraphe en minuscules

$fp = fopen("php://stdin", "r") or die("Impossible de lire la ligne de commande");
while (!feof($fp)) {
    $line = fgets($fp);
    $line = preg_replace_callback(
        '|<p>\s*\w|',
        create_function(
            // Les guillemets simples sont très importants ici
            // ou bien il faut protéger les caractères $ avec \$
            '$matches',
            'return strtolower($matches[0]);'
        ),
        $line
    );
    echo $line;
}
fclose($fp);
?>
```

Voir aussi `preg_replace`, `create_function`, et information à propos de `callback` type.

8.99.15 `preg_replace()` : Rechercher et remplacer par expression rationnelle standard

mixed `preg_replace` (mixed pattern , mixed replacement , mixed subject , int limit , int count)

`preg_replace` analyse subject pour trouver l'expression rationnelle pattern et remplace les résultats par remplacement .

remplacement peut contenir des références de la forme `\n` ou, (depuis PHP 4.0.4) `$n` . Cette dernière forme est recommandée. Ces références seront remplacées par le texte capturé par la n-ième parenthèse capturante du masque. n peut prendre des valeurs de 0 à 99, et `\0` ou `$0` , correspondent au texte de qui satisfait le masque complet. Les parenthèses ouvrantes sont comptées de gauche à droite (en commençant à 1) pour déterminer le numéro de parenthèse capturante.

Lorsque vous travaillez avec un masque de remplacement où une référence arrière est directement suivie par un nombre (i.e.: placer un nombre littéral immédiatement après une référence arrière), vous ne pouvez pas utiliser la syntaxe classique `\1` pour la référence arrière. `\11` , par exemple, sera confus pour la fonction `preg_replace` dans le sens où elle ne saura pas si vous désirez la référence arrière `\1` suivi du nombre 1 ou si vous désirez la référence arrière `\11` suivi de "rien". Dans ce cas, la solution est d'utiliser la syntaxe `\${1}1` . Cela créera une référence arrière isolée `$1` , suivi du nombre littéral 1 .

Si subject est un tableau, alors l'opération sera appliquée à chacun des éléments du tableau, et le tableau sera retourné.

e force `preg_replace` à traiter remplacement comme du code PHP une fois que les substitutions adéquates ont été faites. Conseil : assurez-vous que remplacement est un code PHP valide, car sinon, PHP trouvera une erreur d'analyse dans la ligne contenant la fonction `preg_replace` .

8.99.16 `preg_split()` : Eclate une chaîne par expression rationnelle

array `preg_split` (string pattern , string subject , int limit , int flags)

`preg_split` retourne un tableau contenant les sous-chaînes de subject , séparées par les chaînes qui vérifient pattern .

Si limit est spécifié, alors seules les limit premières sous-chaînes sont retournées et si limit vaut -1, cela signifie en fait "sans limite", ce qui est utile pour passer le paramètre flags .

flags peut être la combinaison des options suivantes (combinées avec l'opérateur |):

`PREG_SPLIT_NO_EMPTY`

Si cette option est activée, seules les sous-chaînes non vides seront retournées par `preg_split` .

`PREG_SPLIT_DELIM_CAPTURE`

Si cette option est activée, les expressions entre parenthèses entre les délimiteurs de masques seront aussi capturées et retournées. Cette option a été ajoutée en PHP 4.0.5.

`PREG_SPLIT_OFFSET_CAPTURE`

Si cette option est activée, pour chaque résultat, la position de celui-ci sera retournée. Notez que cela change la valeur retournée en un tableau où chaque élément est un tableau constitué de la chaîne trouvée à la position 0 et la position de la chaîne dans subject à la position 1 . Cette option est disponible depuis PHP 4.3.0.

Eclatement d'une chaîne de recherche

```
<?php
// scinde la phrase grâce aux virgules et espaces
```

```
// ce qui inclus les " ", \r, \t, \n et \f
$keywords = preg_split("/[\s,]+/", "langage hypertexte, programmation");
?>
```

Scinder une chaîne en caractères

```
<?php
$str = 'string';
$chars = preg_split('//', $str, -1, PREG_SPLIT_NO_EMPTY);
print_r($chars);
?>
```

Scinde une chaîne et capture les positions

```
<?php
$str = 'langage hypertexte, programmation';
$chars = preg_split('/ /', $str, -1, PREG_SPLIT_OFFSET_CAPTURE);
print_r($chars);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
  [0] => Array
  (
    [0] => langage
    [1] => 0
  )
  [1] => Array
  (
    [0] => hypertexte,
    [1] => 8
  )
  [2] => Array
  (
    [0] => programmation
    [1] => 20
  )
)
```

Note

Le paramètre flags a été ajouté en PHP 4 Bêta 3.

Voir aussi [spliti](#) , [split](#) , [implode](#) , [preg_match](#) , [preg_match_all](#) et [preg_replace](#) .

8.100 PDF

8.100.1 Introduction

Les fonctions PDF permettent de créer des fichiers PDF en utilisant la bibliothèque PDFlib créée par [Thomas Merz](#) .

La documentation de cette section est uniquement destinée à être une introduction aux fonctions disponibles, et ne doit pas être considérée comme une référence exhaustive. Reportez-vous à la documentation incluse dans la distribution de PDFlib pour plus de détails et d'explications. Elle fournit une excellente référence des capacités de PDFlib, et contient la documentation la plus à jour.

Toutes les fonctions de la PDFlib et du module PHP ont des noms et des arguments identiques. Vous aurez à comprendre quelques concepts basiques de PDF et PostScript pour l'utiliser efficacement. Toutes les longueurs et coordonnées sont mesurées en points PostScript. Il y a généralement 72 points PostScript pour faire un pouce (2.54 cm), mais cela dépend de la résolution de sortie. Reportez-vous à la documentation PDFlib incluse dans la distribution pour plus de détails sur le système de coordonnées.

Notez que la plupart des fonctions PDF requièrent une ressource pdf object comme premier paramètre. Voyez [les exemples](#) ci-dessous pour plus de détails.

Note
Si vous êtes intéressé par des alternatives gratuites pour générer des PDF, sans passer par des bibliothèques PDF, voyez cette entrée de la FAQ .
Note
Cette extension a été déplacée dans PECL depuis PHP 4.3.9.

8.100.2 Pré-requis

PDFlib est disponible en téléchargement à <http://www.pdfli.com/products/pdfli/index.html> , mais impose l'achat d'une licence pour l'utiliser commercialement. Les bibliothèques [JPEG](#) et [TIFF](#) sont nécessaires pour faire fonctionner cette bibliothèque.

8.100.2.1 Problèmes avec les anciennes versions de PDFlib

Toutes les versions de PHP 4 éditées après le 9 Mars 2000 ne suportent pas les versions de PDFlib plus anciennes que PDFlib 3.0.

PDFlib 3.0 et plus récent est supporté par PHP 3.0.19 et plus récent.

8.100.3 Installation

Cette extension [PECL](#) n'est pas intégrée à PHP. Des informations comme les notes sur les nouvelles versions, les téléchargements, les sources des fichiers, les informations concernant les mainteneurs ainsi qu'un CHANGELOG, peuvent être trouvées ici : <http://pecl.php.net/package/pdfli> .

Pour inclure le support de la PDFlib dans votre PHP < 4.3.9, il faut compiler PHP avec l'option `--with-pdflib[=DIR]` . DIR est le dossier d'installation de PDFlib et, par défaut, il vaut `/usr/local` . De plus, vous pouvez aussi spécifier les dossiers des bibliothèques jpeg, tiff et png que vous voulez utiliser avec la PDFlib. Ceci est optionnel pour PDFlib 4.x. Pour cela, ajoutez les lignes suivantes à votre script de configuration : `--with-jpeg-dir[=DIR]` , `--with-png-dir[=DIR]` et `--with-tiff-dir[=DIR]` .

Lorsque vous utilisez les versions 3.x de PDFlib, vous devez configurer PDFlib avec l'option `--enable-shared-pdflib` .

Depuis PHP 4.3.9, vous devez installer cette extension via [PEAR](#) , en utilisant la commande suivante : `pear install pdflib` .

8.100.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.100.5 Confusion entre les vieilles versions de PDFLib

Depuis le début du support de PDF sous PHP, (commençant avec la version PDFLib 0.6), il y a eu des milliers de modifications dans les API de PDFLib. La plupart de ces modifications ont été suivies par PHP, et parfois même au prix de modifications des API PHP. Depuis la version 3.x, ces API semblent s'être stabilisées, et PHP 4 a adopté cette version comme le minimum nécessaire pour supporter PDF. En conséquence de quoi, un grand nombre de fonctions vont disparaître, ou être remplacées. Le support de PDFLib 0.6 est complètement abandonné. La liste suivante indique quelles sont les fonctions obsolètes en PHP 4.02, et qui devraient être remplacées par de nouvelles versions.

Ancienne fonction	Remplacement
<u>pdf_put_image</u>	Désormais inutile.
<u>pdf_execute_image</u>	Désormais inutile.
<u>pdf_get_annotation</u>	<u>pdf_get_bookmark</u> en utilisant les mêmes paramètres.
<u>pdf_get_font</u>	<u>pdf_get_value</u> en passant "font" comme second paramètre.
<u>pdf_get_fontsize</u>	<u>pdf_get_value</u> en passant "fontsize" comme second paramètre.
<u>pdf_get_fontname</u>	<u>pdf_get_parameter</u> en passant "fontname" comme second paramètre.
<u>pdf_set_info_creator</u>	<u>pdf_set_info</u> en passant "Creator" comme second paramètre.
<u>pdf_set_info_title</u>	<u>pdf_set_info</u> en passant "Title" comme second paramètre.
<u>pdf_set_info_subject</u>	<u>pdf_set_info</u> en passant "Subject" comme second paramètre.
<u>pdf_set_info_author</u>	<u>pdf_set_info</u> en passant "Author" comme second paramètre.
<u>pdf_set_info_keywords</u>	<u>pdf_set_info</u> en passant "Keywords" comme second paramètre.
<u>pdf_set_leading</u>	<u>pdf_set_value</u> en passant "leading" comme second paramètre.
<u>pdf_set_text_rendering</u>	<u>pdf_set_value</u> en passant "textrendering" comme second paramètre.
<u>pdf_set_text_rise</u>	<u>pdf_set_value</u> en passant "textrise" comme second paramètre.
<u>pdf_set_horiz_scaling</u>	<u>pdf_set_value</u> en passant "horizscaling" comme second paramètre.
<u>pdf_set_text_matrix</u>	Désormais abandonnée.
<u>pdf_set_char_spacing</u>	<u>pdf_set_value</u> en passant "charspacing" comme second paramètre.
<u>pdf_set_word_spacing</u>	<u>pdf_set_value</u> en passant "wordspacing" comme second paramètre.
<u>pdf_set_transition</u>	<u>pdf_set_parameter</u> en passant "transition" comme second paramètre.

pdf_open	pdf_new suivie d'un appel à pdf_open_file
pdf_set_font	pdf_findfont suivie d'un appel à pdf_setfont
pdf_set_duration	pdf_set_value en passant "duration" comme second paramètre.
pdf_open_gif	pdf_open_image_file en passant "gif" comme second paramètre.
pdf_open_jpeg	pdf_open_image_file en passant "jpeg" comme second paramètre.
pdf_open_tiff	pdf_open_image_file en passant "tiff" comme second paramètre.
pdf_open_png	pdf_open_image_file en passant "png" comme second paramètre.
pdf_get_image_width	pdf_get_value en passant "imagewidth" comme second paramètre et l'image en troisième paramètre.
pdf_get_image_height	pdf_get_value en passant "imageheight" comme second paramètre et l'image en troisième paramètre.

8.100.6 Exemples

La plupart des fonctions sont simples d'emploi. Le plus difficile est probablement de créer un fichier PDF simple. L'exemple suivant devrait vous mettre sur les rails. Il crée un fichier test.pdf d'une page. La page contient du texte "Times Roman outlined", d'une taille de 30pt. Le texte est aussi souligné.

Création d'un document PDF avec PDFlib

```
<?php
$pdf = pdf_new();
pdf_open_file($pdf, "test.pdf");
pdf_set_info($pdf, "Author", "Uwe Steinmann");
pdf_set_info($pdf, "Title", "Test for PHP wrapper of PDFlib 2.0");
pdf_set_info($pdf, "Creator", "See Author");
pdf_set_info($pdf, "Subject", "Testing");
pdf_begin_page($pdf, 595, 842);
pdf_add_outline($pdf, "Page 1");
$font = pdf_findfont($pdf, "Times New Roman", "winansi", 1);
pdf_setfont($pdf, $font, 10);
pdf_set_value($pdf, "textrendering", 1);
pdf_show_xy($pdf, "Times Roman outlined", 50, 750);
pdf_moveto($pdf, 50, 740);
pdf_lineto($pdf, 330, 740);
pdf_stroke($pdf);
pdf_end_page($pdf);
pdf_close($pdf);
pdf_delete($pdf);
echo '<a href="getpdf.php">terminé</a>';
?>
```

Le script getpdf.php ne fait que retourner le document PDF.

Affichage d'un PDF précalculé

```
<?php
$flen = filesize($filename);
header("Content-type: application/pdf");
header("Content-Length: $flen");
header("Content-Disposition: inline; filename=foo.pdf");
readfile($filename);
?>
```

La distribution de la PDFlib contient des exemples plus complexes, qui créent une page avec une horloge à aiguille. Ici, nous utilisons la création de pages en mémoire, pour éviter l'utilisation de fichiers temporaires. L'exemple a été adapté à PHP. (Cet exemple est disponible dans la documentation [CLibPDF](#).)

Exemple pdfclock de PDFlib

```

<?php
$radius = 200;
$margin = 20;
$pagecount = 10;

$pdf = pdf_new();

if (!pdf_open_file($pdf, "")) {
    echo error;
    exit;
};

pdf_set_parameter($pdf, "warning", "true");

pdf_set_info($pdf, "Creator", "pdf_clock.php");
pdf_set_info($pdf, "Author", "Uwe Steinmann");
pdf_set_info($pdf, "Title", "Analog Clock");

while ($pagecount-- > 0) {
    pdf_begin_page($pdf, 2 * ($radius + $margin), 2 * ($radius + $margin));

    pdf_set_parameter($pdf, "transition", "wipe");
    pdf_set_value($pdf, "duration", 0.5);

    pdf_translate($pdf, $radius + $margin, $radius + $margin);
    pdf_save($pdf);
    pdf_setrgbcolor($pdf, 0.0, 0.0, 1.0);

    /* les marques des minutes */
    pdf_setlinewidth($pdf, 2.0);
    for ($alpha = 0; $alpha < 360; $alpha += 6) {
        pdf_rotate($pdf, 6.0);
        pdf_moveto($pdf, $radius, 0.0);
        pdf_lineto($pdf, $radius-$margin/3, 0.0);
        pdf_stroke($pdf);
    }

    pdf_restore($pdf);
    pdf_save($pdf);

    /* les marques des 5 minutes */
    pdf_setlinewidth($pdf, 3.0);
    for ($alpha = 0; $alpha < 360; $alpha += 30) {
        pdf_rotate($pdf, 30.0);
        pdf_moveto($pdf, $radius, 0.0);
        pdf_lineto($pdf, $radius-$margin, 0.0);
        pdf_stroke($pdf);
    }

    $ltime = getdate();

    /* l'aiguille des heures */
    pdf_save($pdf);
    pdf_rotate($pdf, -((($ltime['minutes']/60.0)+$ltime['hours']-3.0)*30.0));
    pdf_moveto($pdf, -$radius/10, -$radius/20);
    pdf_lineto($pdf, $radius/2, 0.0);
    pdf_lineto($pdf, -$radius/10, $radius/20);
    pdf_closepath($pdf);
    pdf_fill($pdf);
    pdf_restore($pdf);

    /* l'aiguille des minutes */
    pdf_save($pdf);
    pdf_rotate($pdf, -((($ltime['seconds']/60.0)+$ltime['minutes']-15.0)*6.0));
    pdf_moveto($pdf, -$radius/10, -$radius/20);

```

```

pdf_lineto($pdf, $radius * 0.8, 0.0);
pdf_lineto($pdf, -$radius/10, $radius/20);
pdf_closepath($pdf);
pdf_fill($pdf);
pdf_restore($pdf);

/* l'aiguille des secondes */
pdf_setrgbcolor($pdf, 1.0, 0.0, 0.0);
pdf_setlinewidth($pdf, 2);
pdf_save($pdf);
pdf_rotate($pdf, -(($ltime['seconds'] - 15.0) * 6.0));
pdf_moveto($pdf, -$radius/5, 0.0);
pdf_lineto($pdf, $radius, 0.0);
pdf_stroke($pdf);
pdf_restore($pdf);

/* Dessin d'un petit cercle dans le centre */
pdf_circle($pdf, 0, 0, $radius/30);
pdf_fill($pdf);

pdf_restore($pdf);

pdf_end_page($pdf);

// pour voir la différence
sleep(1);
}

pdf_close($pdf);

$buf = pdf_get_buffer($pdf);
$len = strlen($buf);

header("Content-type: application/pdf");
header("Content-Length: $len");
header("Content-Disposition: inline; filename=foo.pdf");
echo $buf;

pdf_delete($pdf);
?>

```

8.100.7 Voir aussi

Note

Une alternative au module module PDFlib est le module de [FastIO](#) : ClibPDF. Voyez la section [ClibPDF](#) pour plus de détails. Notez que [ClibPDF](#) dispose d'une API différente de PDFlib.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Confusion entre les vieilles versions de PDFLib](#)
- [Exemples](#)
- [Voir aussi](#)
- [pdf_add_annotation](#)
- [pdf_add_bookmark](#)
- [pdf_add_launchlink](#)
- [pdf_add_loccallink](#)
- [pdf_add_note](#)
- [pdf_add_outline](#)

- [pdf_add_pdflink](#)
- [pdf_add_thumbnail](#)
- [pdf_add_weblink](#)
- [pdf_arc](#)
- [pdf_arcn](#)
- [pdf_attach_file](#)
- [pdf_begin_page](#)
- [pdf_begin_pattern](#)
- [pdf_begin_template](#)
- [pdf_circle](#)
- [pdf_clip](#)
- [pdf_close_image](#)
- [pdf_close_pdi_page](#)
- [pdf_close_pdi](#)
- [pdf_close](#)
- [pdf_closepath_fill_stroke](#)
- [pdf_closepath_stroke](#)
- [pdf_closepath](#)
- [pdf_concat](#)
- [pdf_continue_text](#)
- [pdf_curveto](#)
- [pdf_delete](#)
- [pdf_end_page](#)
- [pdf_end_pattern](#)
- [pdf_end_template](#)
- [pdf_endpath](#)
- [pdf_fill_stroke](#)
- [pdf_fill](#)
- [pdf_findfont](#)
- [pdf_get_buffer](#)
- [pdf_get_font](#)
- [pdf_get_fontname](#)
- [pdf_get_fontsize](#)
- [pdf_get_image_height](#)
- [pdf_get_image_width](#)
- [pdf_get_majorversion](#)
- [pdf_get_minorversion](#)
- [pdf_get_parameter](#)
- [pdf_get_pdi_parameter](#)
- [pdf_get_pdi_value](#)
- [pdf_get_value](#)
- [pdf_initgraphics](#)
- [pdf_lineto](#)
- [pdf_makespotcolor](#)
- [pdf_moveto](#)
- [pdf_new](#)
- [pdf_open_ccitt](#)
- [pdf_open_file](#)
- [pdf_open_gif](#)
- [pdf_open_image_file](#)
- [pdf_open_image](#)
- [pdf_open_jpeg](#)
- [pdf_open_memory_image](#)
- [pdf_open_pdi_page](#)

- [pdf_open_pdi](#)
- [pdf_open_png](#)
- [pdf_open_tiff](#)
- [pdf_open](#)
- [pdf_place_image](#)
- [pdf_place_pdi_page](#)
- [pdf_rect](#)
- [pdf_restore](#)
- [pdf_rotate](#)
- [pdf_save](#)
- [pdf_scale](#)
- [pdf_set_border_color](#)
- [pdf_set_border_dash](#)
- [pdf_set_border_style](#)
- [pdf_set_char_spacing](#)
- [pdf_set_duration](#)
- [pdf_set_font](#)
- [pdf_set_horiz_scaling](#)
- [pdf_set_info_author](#)
- [pdf_set_info_creator](#)
- [pdf_set_info_keywords](#)
- [pdf_set_info_subject](#)
- [pdf_set_info_title](#)
- [pdf_set_info](#)
- [pdf_set_leading](#)
- [pdf_set_parameter](#)
- [pdf_set_text_matrix](#)
- [pdf_set_text_pos](#)
- [pdf_set_text_rendering](#)
- [pdf_set_text_rise](#)
- [pdf_set_value](#)
- [pdf_set_word_spacing](#)
- [pdf_setcolor](#)
- [pdf_setdash](#)
- [pdf_setflat](#)
- [pdf_setfont](#)
- [pdf_setgray_fill](#)
- [pdf_setgray_stroke](#)
- [pdf_setgray](#)
- [pdf_setlinecap](#)
- [pdf_setlinejoin](#)
- [pdf_setlinewidth](#)
- [pdf_setmatrix](#)
- [pdf_setmiterlimit](#)
- [pdf_setpolydash](#)
- [pdf_setrgbcolor_fill](#)
- [pdf_setrgbcolor_stroke](#)
- [pdf_setrgbcolor](#)
- [pdf_show_boxed](#)
- [pdf_show_xy](#)
- [pdf_show](#)
- [pdf_skew](#)
- [pdf_stringwidth](#)
- [pdf_stroke](#)

- [pdf_translate](#)

8.100.9 pdf_add_bookmark() : Ajoute un signet dans la page courante

int **pdf_add_bookmark** (resource pdfdoc , string text , int parent , int open)

[pdf_add_bookmark](#) ajoute un signet imbriqué dans la page courante, sous parent , ou bien dans le signet global si parent est égal à 0. [pdf_add_bookmark](#) retourne une ressource de signet, qui peut servir de paramètre parent . Si open vaut 1, le signet fils sera visible. Si open vaut 0, le signet fils sera invisible. Les paramètres parent et open sont optionnels avant PHP 4.3.5 ou avec PDFlib inférieur à 5.

Exemple avec [pdf_add_bookmark](#)

```
<?php
// création d'un nouveau PDF

$pdf = pdf_new();
pdf_open_file($pdf);
pdf_set_info($pdf, "Author", "Bob Nijman");

// on commence une nouvelle page
pdf_begin_page($pdf, 300, 300);

// ajout d'un signet en haut de la page
$bookmark = pdf_add_bookmark($pdf, "People");

// ajout d'un bookmark nommé
pdf_add_bookmark($pdf, "Rasmus", $bookmark);

// et un peu de texte
pdf_set_font($pdf, "Helvetica", 20, "host");
$text = "C'est la page de R";
$width = pdf_stringwidth($pdf, $text);
pdf_set_text_pos($pdf, (300-$width)/2, 100);
pdf_show($pdf, $text);

// ferme la page et le PDF
pdf_end_page($pdf);
pdf_close($pdf);

?>
```

8.100.10 pdf_add_launchlink() : Ajoute une annotation de lancement dans la page PDF courante

bool **pdf_add_launchlink** (resource pdfdoc , float llx , float lly , float urx , float ury , string filename)

[pdf_add_launchlink](#) ajoute une annotation de lancement de fichier (type arbitraire de fichiers). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [pdf_add_loclink](#) .

8.100.11 pdf_add_locallink() : Ajoute une annotation de lien dans la page PDF courante

bool **pdf_add_locallink** (resource pdfdoc , float lowerleftx , float lowerlefty , float upperrightx , float upperrighty , int page , string dest)

pdf_add_locallink ajoute une annotation de lien dans la page courante. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

dest est le zoom de la page de destination. Il peut valoir retain , fitpage , fitwidth , fiteight ou fitbbox .

Voir aussi pdf_add_launchlink .

8.100.12 pdf_add_note() : Ajoute une annotation dans la page PDF courante

bool **pdf_add_note** (resource pdfdoc , float llx , float lly , float urx , float ury , string contents , string title , string icon , int open)

pdf_add_note ajoute une annotation dans la page courante. Le paramètre icon peut prendre l'une des valeurs suivantes : " comment " , " insert " , " note " , " paragraph " , " newparagraph " , " key " et " help " . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Ajout d'une note à un document PDF avec la PDFlib

```
<?php
$pdfdoc = pdf_new();
if (!pdf_open_file($pdfdoc, '')) {
    echo 'erreur!';
    exit;
}
pdf_begin_page($pdfdoc, 300, 300);
if ($font = pdf_findfont($pdfdoc, 'Times-Roman', 'host', 0)) {
    pdf_setfont($pdfdoc, $font, 10);
}
pdf_set_value($pdfdoc, 'textrendering', 0);
pdf_add_note($pdfdoc, 100, 100, 180, 40, 'Contenu de la note', 'Titre', 'aide', 0);
pdf_show_xy($pdfdoc, 'Example', 100, 100);
pdf_end_page($pdfdoc);
pdf_close($pdfdoc);
$data = pdf_get_buffer($pdfdoc);
header('Content-type: application/pdf');
header('Content-disposition: inline; filename=example.pdf');
header('Content-length: '.strlen($data));
echo $data;
?>
```

8.100.13 pdf_add_outline() : Obsolète : ajoute un signet dans la page courante

pdf_add_outline a été remplacé par pdf_add_bookmark .

8.100.14 pdf_add_pdflink() : Ajoute une annotation sur un lien vers un fichier dans la page PDF courante

bool **pdf_add_pdflink** (resource pdfdoc , float bottom_left_x , float bottom_left_y , float up_right_x , float up_right_y , string filename , int page , string dest)

pdf_add_pdflink ajoute une annotation sur un lien vers un fichier dans la page courante. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_add_loccallink et pdf_add_weblink .

8.100.15 pdf_add_thumbnail() : Ajoute une miniature sur la page PDF courante

bool **pdf_add_thumbnail** (resource pdfdoc , int image)

pdf_add_thumbnail ajoute une miniature sur la page courante. La miniature ne peut être plus large ou plus haute que 106 pixels. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_open_image , pdf_open_image_file et pdf_open_memory_image .

8.100.16 pdf_add_weblink() : Ajoute un lien web sur la page PDF courante

bool **pdf_add_weblink** (resource pdfdoc , float lowerleftx , float lowerlefty , float upperrightx , float upperrighty , string url)

pdf_add_weblink ajoute un lien web url sur la page courante. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.17 pdf_arc() : Dessine un arc de cercle PDF dans le sens antihoraire

bool **pdf_arc** (resource pdfdoc , float x , float y , float r , float alpha , float beta)

pdf_arc dessine un arc de cercle dans le sens antihoraire, dont le centre est le point de coordonnées (x , y) et de rayon r , depuis l'angle alpha jusqu'à beta , exprimés en degrés. Le schéma réel du cercle est exécuté par la prochaine opération de dessin. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec pdf_arcn

```
<?php
// Préparation du document
$pdf = pdf_new();
pdf_open_file($pdf, "");
pdf_begin_page($pdf, 595, 842);

// un arc décrit
pdf_arc($pdf, 200, 700, 100, 0, 90);
pdf_stroke($pdf);
```

```
// un arc rempli
pdf_arc($pdf, 200, 700, 50, 0, 90);
pdf_fill($pdf);

// les deux
pdf_setcolor($pdf, "fill", "gray", 0.8);
pdf_arc($pdf, 400, 700, 50, 0, 90);
pdf_fill_stroke($pdf);

// Fin du document
pdf_end_page($pdf);
pdf_close($pdf);

header("Content-type: application/pdf");
echo pdf_get_buffer($pdf);

pdf_delete($pdf);
?>
```

Voir aussi [pdf_arcn](#) , [pdf_circle](#) , [pdf_stroke](#) , [pdf_fill](#) et [pdf_fill_stroke](#) .

8.100.18 pdf_arcn() : Dessine un arc de cercle dans le sens horaire

bool **pdf_arcn** (resource pdfdoc , float x , float y , float r , float alpha , float beta)

pdf_arc dessine un arc de cercle dans le sens horaire dont le centre est le point de coordonnées (x , y) et de rayon r , depuis l'angle alpha jusqu'à beta , exprimés en degrés. Le schéma réel du cercle est exécuté par la prochaine opération de dessin. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec [pdf_arcn](#)

```
<?php

// Préparation du document
$pdf = pdf_new();
pdf_open_file($pdf, "");
pdf_begin_page($pdf, 595, 842);

// un arc décrit
pdf_arcn($pdf, 200, 700, 100, 0, 90);
pdf_stroke($pdf);

// un arc rempli
pdf_arcn($pdf, 200, 700, 50, 0, 90);
pdf_fill($pdf);

// les deux
pdf_setcolor($pdf, "fill", "gray", 0.8);
pdf_arcn($pdf, 400, 700, 50, 0, 90);
pdf_fill_stroke($pdf);

// Fin du document
pdf_end_page($pdf);
pdf_close($pdf);

header("Content-type: application/pdf");
echo pdf_get_buffer($pdf);

pdf_delete($pdf);
?>
```

Voir aussi [pdf_arc](#) , [pdf_circle](#) , [pdf_stroke](#) , [pdf_fill](#) et [pdf_fillstroke](#) .

8.100.19 pdf_attach_file() : Ajoute un fichier attaché à la page PDF

bool **pdf_attach_file** (resource pdfdoc , float llx , float lly , float urx , float ury , string filename , string description , string author , string mimetype , string icon)

pdf_attach_file ajoute un fichier attaché à la page courante. icon peut prendre l'une des valeurs suivantes : graph , paperclip , pushpin ou tag . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note
Seule la version complète d'Acrobat sera capable d'afficher les attachements. Toutes les autres versions d'Acrobat n'afficheront qu'un point d'interrogation.

8.100.20 pdf_begin_page() : Initialise une nouvelle page de document PDF

bool **pdf_begin_page** (resource pdfdoc , float width , float height)

pdf_begin_page ajoute une nouvelle page au document. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. Les paramètres width et height sont spécifiés en points, qui représentent 1/72 d'un pouce (2.54 cm).

nom	taille
A0	2380 par 3368
A1	1684 par 2380
A2	1190 par 1684
A3	842 par 1190
A4	595 par 842
A5	421 par 595
A6	297 par 421
B5	501 par 709
letter (8.5" par 11")	612 par 792
legal (8.5" par 14")	612 par 1008
ledger (17" par 11")	1224 par 792
11" par 17"	792 par 1224

Voir aussi [pdf_end_page](#) .

8.100.21 pdf_begin_pattern() : Initialise un nouveau pattern PDF

int **pdf_begin_pattern** (resource pdfdoc , float width , float height , float xstep , float ystep , int painttype)

pdf_begin_pattern initialise un nouveau pattern, et retourne une ressource. width et height définissent le cadre du pattern. xstep et ystep représentent les offsets des répétitions. painttype =1 indique que le pattern dispose de ses propres couleurs, tandis que la valeur 2 indique que la couleur courante est utilisée lorsque le pattern est appliqué.

Voir aussi [pdf_end_pattern](#) .

8.100.22 pdf_begin_template() : Initialise un nouveau template PDF

int **pdf_begin_template** (resource pdfdoc , float width , float height)

[pdf_begin_template](#) initialise un nouveau template.

8.100.23 pdf_circle() : Dessine un cercle dans un document PDF

bool **pdf_circle** (resource pdfdoc , float x , float y , float r)

[pdf_circle](#) dessine un cercle de centre (x , y) et de rayon r . Le schéma réel du cercle est exécuté par la prochaine opération de dessin.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec [pdf_circle](#)

```
<?php
// Préparation du document
$pdf = pdf_new();
pdf_open_file($pdf, "");
pdf_begin_page($pdf, 595, 842);

// un cercle décrit
pdf_circle($pdf, 200, 700, 100);
pdf_stroke($pdf);

// un cercle rempli
pdf_circle($pdf, 200, 700, 50);
pdf_fill($pdf);

// les deux
pdf_setcolor($pdf, "fill", "gray", 0.3);
pdf_circle($pdf, 400, 700, 50);
pdf_fill_stroke($pdf);

// Fin du document
pdf_end_page($pdf);
pdf_close($pdf);

header("Content-type: application/pdf");
echo pdf_get_buffer($pdf);

pdf_delete($pdf);
?>
```

Voir aussi [pdf_arc](#) et [pdf_arcn](#) .

8.100.24 pdf_clip() : Modifie le chemin de clipping PDF

bool **pdf_clip** (resource pdfdoc)

[pdf_clip](#) utilise le chemin courant comme chemin de clipping. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.25 `pdf_close_image()` : Ferme une image dans un document PDF

`void pdf_close_image (resource pdfdoc , int image)`

`pdf_close_image` ferme l'image `image` , ouverte avec l'une des fonctions images : `pdf_open_image` , `pdf_open_gif` , `pdf_open_jpg` , `pdf_open_png` , `pdf_open_image_file` , `pdf_open_tiff` et `pdf_open_memory_image` .

8.100.26 `pdf_close_pdi_page()` : Ferme la page PDF

`bool pdf_close_pdi_page (resource pdfdoc , int pagehandle)`

`pdf_close_pdi_page` ferme la page, et libère toutes les ressources liées à celle-ci. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.27 `pdf_close_pdi()` : Ferme le fichier PDF d'entrée

`bool pdf_close_pdi (resource pdfdoc , int dochandle)`

`pdf_close_pdi` ferme toutes les pages ouvertes, et referme le document PDF utilisé en entrée. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `pdf_open_pdi` .

8.100.28 `pdf_close()` : Ferme le fichier PDF

`bool pdf_close (resource pdfdoc)`

`pdf_close` ferme le fichier PDF généré, et libère toutes les ressources qui y était rattachées. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `pdf_new` .

8.100.29 `pdf_closepath_fill_stroke()` : Termine le chemin, dessine les bords et remplit la forme

`bool pdf_closepath_fill_stroke (resource pdfdoc)`

`pdf_closepath_fill_stroke` termine le chemin, dessine les bords et remplit la forme. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi `pdf_closepath` et `pdf_closepath_stroke` .

8.100.30 `pdf_closepath_stroke()` : Termine le chemin et dessine les bords

`bool pdf_closepath_stroke (resource pdfdoc)`

pdf_closepath_stroke termine le chemin et dessine les bords. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_closepath et pdf_closepath_fil_stroke .

8.100.31 pdf_closepath() : Termine le chemin PDF courant

bool **pdf_closepath** (resource pdfdoc)

pdf_closepath termine le chemin courant. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_closepath_stroke et pdf_closepath_fil_stroke .

8.100.32 pdf_concat() : Concatène une matrice avec le CTM

bool **pdf_concat** (resource pdfdoc , float a , float b , float c , float d , float e , float f)

pdf_concat concatène une matrice avec le CTM. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.33 pdf_continue_text() : Affiche du texte à la prochaine ligne PDF

bool **pdf_continue_text** (resource pdfdoc , string text)

pdf_continue_text affiche le texte text à la prochaine ligne. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.34 pdf_curveto() : Dessine une courbe de Bezier

bool **pdf_curveto** (resource pdfdoc , float x1 , float y1 , float x2 , float y2 , float x3 , float y3)

pdf_curveto dessine une courbe de Bezier à partir du point courant, et en utilisant trois points de contrôle. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.35 pdf_delete() : Efface un objet PDF

bool **pdf_delete** (resource pdfdoc)

pdf_delete efface l'objet PDF pdf_object , et libère toutes les ressources. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_new .

8.100.36 pdf_end_page() : Termine la page PDF courante

bool **pdf_end_page** (resource pdfdoc)

pdf_end_page termine la page courante. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_begin_page .

8.100.37 pdf_end_pattern() : Termine le pattern PDF

bool **pdf_end_pattern** (resource pdfdoc)

pdf_end_pattern termine la définition du pattern. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_begin_pattern .

8.100.38 pdf_end_template() : Termine le template PDF

bool **pdf_end_template** (resource pdfdoc)

pdf_end_template termine la définition du template. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.39 pdf_endpath() : Obsolète : termine le chemin courant

Obsolète. Utilisez une des fonctions de pinceau, remplissage ou de clip.

Voir aussi pdf_stroke , pdf_clip et pdf_closepath_fill_stroke .

8.100.40 pdf_fill_stroke() : Remplit et passe le pinceau sur le chemin PDF courant

bool **pdf_fill_stroke** (resource pdfdoc)

pdf_fill_stroke remplit le chemin courant avec la couleur courante, en appliquant le pinceau courant (stroke). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_setcolor .

8.100.41 pdf_fill() : Remplit le chemin PDF courant avec la couleur courante

bool **pdf_fill** (resource pdfdoc)

[pdf_fill_stroke](#) remplit le chemin courant avec la couleur courante. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [pdf_setcolor](#) .

8.100.42 pdf_findfont() : Prépare une police pour utilisation ultérieure avec pdf_setfont

int **pdf_findfont** (resource pdfdoc , string fontname , string encoding , int embed)

[pdf_findfont](#) prépare la police fontname pour utilisation ultérieure avec [pdf_setfont](#) . Les dimensions seront chargées et, si embed est non nul, le fichier de police sera vérifié mais pas utilisé. encoding peut prendre une des valeurs suivantes : builtin , macroman , winansi , host , ou bien un encodage défini par l'utilisateur, ou encore le nom d'une CMap. Le paramètre embed est optionnel avant PHP 4.3.5 ou avec PDFlib inférieur à 5.

[pdf_findfont](#) retourne une ressource de police, ou bien FALSE en cas d'erreur.

Exemple avec [pdf_findfont](#)

```
<?php
$font = pdf_findfont($pdf, "Times New Roman", "winansi", 1);
if ($font) {
    pdf_setfont($pdf, $font, 10);
}
?>
```

8.100.43 pdf_get_buffer() : Lit le tampon contenant le fichier PDF généré

string **pdf_get_buffer** (resource pdfdoc)

[pdf_get_buffer](#) lit le contenu du tampon PDF. Le résultat doit être utilisé par le client avant d'appeler toute autre fonction PDFlib.

8.100.44 pdf_get_font() : Obsolète: charge une police

Obsolète.

Voir [pdf_get_value](#) .

8.100.45 pdf_get_fontname() : Obsolète : lit le nom de la police

Obsolète.

Voir [pdf_get_parameter](#) .

8.100.46 pdf_get_fontsize() : Obsolète : gère les polices

Obsolète.

Voir [pdf_get_value](#) .

8.100.47 pdf_get_image_height() : Obsolète : retourne la hauteur d'une image

Obsolète

Voir [pdf_get_value](#) .

8.100.48 pdf_get_image_width() : Obsolète : retourne la largeur d'une image

Obsolète

Voir [pdf_get_value](#) .

8.100.49 pdf_get_majorversion() : Retourne le numéro de version majeur de PDFlib

int **pdf_get_majorversion** (void)

[pdf_get_majorversion](#) retourne le numéro de version majeur de PDFlib.

Voir aussi [pdf_get_minorversion](#) .

8.100.50 pdf_get_minorversion() : Retourne le numéro de version mineure de PDFlib

int **pdf_get_minorversion** (void)

[pdf_get_majorversion](#) retourne le numéro de version mineure de PDFlib.

Voir aussi [pdf_get_majorversion](#) .

8.100.51 pdf_get_parameter() : Lit certains paramètres

string **pdf_get_parameter** (resource pdfdoc , string key , float modifier)

[pdf_get_parameter](#) lit certains paramètres au format textuel. Le paramètre modifier est optionnel avant PHP 4.3.5 ou avec PDFlib inférieur à 5.

8.100.52 pdf_get_pdi_parameter() : Lit des paramètres textuels dans le document PDI

string **pdf_get_pdi_parameter** (resource pdfdoc , string key , int document , int page , int index)

pdf_get_pdi_parameter lit des paramètres textuels dans le document PDI document .

8.100.53 pdf_get_pdi_value() : Lit des paramètres numériques dans le document PDF d'entrée

float **pdf_get_pdi_value** (resource pdfdoc , string key , int doc , int page , int index)

pdf_get_pdi_value lit des paramètres numériques dans le document PDF d'entrée. Les paramètres lus sont de type numérique.

8.100.54 pdf_get_value() : Lit certains paramètres numériques

float **pdf_get_value** (resource pdfdoc , string key , float modifier)

pdf_get_value retourne la valeur du paramètre numérique du document pdf_object , identifié par key . Le paramètre modifier est optionnel avant PHP 4.3.5 ou avec PDFlib inférieur à 5.

8.100.55 pdf_initgraphics() : Remet à zéro l'environnement graphique PDF

bool **pdf_initgraphics** (resource pdfdoc)

pdf_initgraphics remet à leur valeur initiale toutes les couleurs et les paramètres graphiques. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.56 pdf_lineto() : Dessine une ligne PDF

bool **pdf_lineto** (resource pdfdoc , float x , float y)

pdf_lineto dessine une ligne entre le point courant et le point de coordonnées (x , y). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.57 pdf_makespotcolor() : Place un point de couleur PDF

int **pdf_makespotcolor** (resource pdfdoc , string spotname)

pdf_makespotcolor place un point de couleur, ayant le nom de spotname et la couleur courante. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_setcolor .

8.100.58 pdf_moveto() : Place le point courant PDF

bool **pdf_moveto** (resource pdfdoc , float x , float y)

pdf_moveto place le point courant aux coordonnées (x , y). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Le point courant graphique et le point courant de texte sont différents. Voyez pdf_set_text_pos pour déplacer le point courant de texte.

8.100.59 pdf_new() : Crée un nouvel objet PDF

resource **pdf_new** ()

pdf_new crée un nouvel objet PDF, en utilisant le système de gestion d'erreurs et celui de gestion de la mémoire.

Voir aussi pdf_close .

8.100.60 pdf_open_ccitt() : Ouvre une image contenant des données brutes CCITT

int **pdf_open_ccitt** (resource pdfdoc , string filename , int width , int height , int BitReverse , int k , int BlackIs1)

pdf_open_ccitt ouvre une image brute CCITT.

8.100.61 pdf_open_file() : Ouvre un nouveau fichier PDF

bool **pdf_open_file** (resource pdfdoc , string filename)

pdf_open_file crée un nouveau fichier PDF dans le fichier filename . Si filename est vide, le document PDF sera envoyé en mémoire. Le résultat pourra alors être lu avec la fonction pdf_get_buffer . Le paramètre optionnel filename est optionnel avant PHP 4.3.5 ou avec PDFlib inférieur à 5. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

L'exemple suivant montre comment créer un document PDF en mémoire et comment l'afficher correctement.

Création d'un document PDF en mémoire

```
<?php
$pdf = pdf_new();

pdf_open_file($pdf);
pdf_begin_page($pdf, 595, 842);
pdf_set_font($pdf, "Times-Roman", 30, "host");
pdf_set_value($pdf, "textrendering", 1);
pdf_show_xy($pdf, "Un document PDF a été créé dans la mémoire !", 50, 750);
pdf_end_page($pdf);
pdf_close($pdf);
```

```
$data = pdf_get_buffer($pdf);

header("Content-type: application/pdf");
header("Content-disposition: inline; filename=test.pdf");
header("Content-length: " . strlen($data));

echo $data;

?>
```

8.100.62 pdf_open_gif() : Obsolète : ouvre une image GIF

Obsolète.

Voir [pdf_open_image](#) .

8.100.63 pdf_open_image_file() : Lit une image depuis un fichier

int **pdf_open_image_file** (resource pdfdoc , string imagetype , string filename , string stringparam , int intparam)

[pdf_open_image_file](#) ouvre une image depuis le fichier filename . Les formats supportés sont jpeg , tiff , gif , et png . stringparam vaut soit la chaîne vide, mask , masked ou page . intparam vaut soit 0, soit la ressource d'image appliquée au masque, soit la ressource de page. Les paramètres stringparam et intparam sont optionnels avant PHP 4.3.5 ou avec PDFlib inférieur à 5.

8.100.64 pdf_open_image() : Ouvre une image

int **pdf_open_image** (resource pdfdoc , string imagetype , string source , string data , int length , int width , int height , int components , int bpc , string params)

[pdf_open_image](#) permet d'accéder à des images de différents formats. Les formats supportés sont : jpeg , ccitt , raw . Les sources supportées sont memory , fileref , url . length est uniquement utilisé si imagetype vaut raw . params est uniquement utilisé si imagetype vaut ccitt .

8.100.65 pdf_open_jpeg() : Obsolète : ouvre une image JPEG

Obsolète.

Voir [pdf_open_image](#) .

8.100.66 pdf_open_memory_image() : Ouvre une image créée en mémoire par PHP

int **pdf_open_memory_image** (resource pdfdoc , resource image)

[pdf_open_memory_image](#) prend l'image image créée par PHP avec les fonctions d'images, et l'intègre dans l'objet PDF pdf_object . Cette fonction retourne une ressource d'image PDF.

Inclure une image PHP en PDF

```
<?php
$im = ImageCreate(100, 100);
$col = ImageColorAllocate($im, 80, 45, 190);
ImageFill($im, 10, 10, $col);
$pim = pdf_open_memory_image($pdf, $im);
ImageDestroy($im);
pdf_place_image($pdf, $pim, 100, 100, 1);
pdf_close_image($pdf, $pim);
?>
```

Voir aussi [pdf_close_image](#) et [pdf_place_image](#) .

8.100.67 pdf_open_pdi_page() : Prépare une page

int **pdf_open_pdi_page** (resource pdfdoc , int dochandle , int pagenumber , string pagelabel)

[pdf_open_pdi_page](#) prépare une page pour utilisation ultérieure avec [pdf_place_pdi_page](#) .

8.100.68 pdf_open_pdi() : Ouvre un fichier PDF

int **pdf_open_pdi** (resource pdfdoc , string filename , string stringparam , int intparam)

[pdf_open_pdi](#) ouvre un fichier PDF existant, pour utilisation ultérieure.

Voir aussi [pdf_close_pdi](#) .

8.100.69 pdf_open_png() : Obsolète : ouvre une image PNG

Obsolète.

Voir [pdf_open_image](#) .

8.100.70 pdf_open_tiff() : Obsolète : ouvre une image TIFF

Obslète.

Voir [pdf_open_image](#) .

8.100.71 pdf_open() : Obsolète : ouvre un nouvel objet PDF

Obsolète.

Voir [pdf_new](#) avec [pdf_open_file](#) .

8.100.72 pdf_place_image() : Place une image dans la page

bool **pdf_place_image** (resource pdfdoc , int image , float x , float y , float scale)

pdf_place_image place l'image image , dans la page courante, au point de coordonnées (x , y), et la met à l'échelle scale . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.73 pdf_place_pdi_page() : Place une page dans le document

bool **pdf_place_pdi_page** (resource pdfdoc , int page , float x , float y , float sx , float sy)

pdf_place_pdi_page place une page PDI, dans la page courante, aux coordonnées (x , y), et la met à l'échelle. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.74 pdf_rect() : Dessine un rectangle

bool **pdf_rect** (resource pdfdoc , float x , float y , float width , float height)

pdf_rect dessine un rectangle de coin inférieur gauche aux coordonnées (x , y), et avec la largeur width et la hauteur height . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.75 pdf_restore() : Rétablit l'ancien environnement graphique PDF

bool **pdf_restore** (resource pdfdoc)

pdf_restore rétablit l'ancien environnement graphique, sauvé avec pdf_save . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.76 pdf_rotate() : Configure la rotation

bool **pdf_rotate** (resource pdfdoc , float phi)

pdf_rotate fait pivoter le système de phi degrés. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.77 pdf_save() : Sauve l'environnement graphique courant

bool **pdf_save** (resource pdfdoc)

pdf_save sauve l'environnement graphique courant. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.78 pdf_scale() : Configure l'échelle du document

bool **pdf_scale** (resource pdfdoc , float x_scale , float y_scale)

[pdf_scale](#) configure l'échelle du document en coordonnées horizontales et verticales. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.79 pdf_set_border_color() : Configure la couleur des bords autour des liens et annotations

bool **pdf_set_border_color** (resource pdfdoc , float red , float green , float blue)

[pdf_set_border_color](#) configure la couleur des bords autour des liens et annotations. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.80 pdf_set_border_dash() : Configure le style des lignes autour des liens et annotations

bool **pdf_set_border_dash** (resource pdfdoc , float black , float white)

[pdf_set_border_dash](#) configure le style des lignes autour des liens. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [pdf_setdash](#) .

8.100.81 pdf_set_border_style() : Choisit le style de bord autour des liens et annotations

bool **pdf_set_border_style** (resource pdfdoc , string style , float width)

[pdf_set_border_style](#) configure le style des bords de tous les liens et annotations. style peut être solid ou dashed . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.82 pdf_set_char_spacing() : Obsolète : configure l'espacement de caractères

Obsolète.

Voir [pdf_set_value](#) .

8.100.83 pdf_set_duration() : Obsolète : configure la durée entre deux pages

Obsolète.

Voir [pdf_set_value](#) .

8.100.84 pdf_set_font() : Obsolète : sélectionne la taille et le nom d'une police

Obsolète.

Voir [pdf_findfont](#) et [pdf_setfont](#) .

8.100.85 pdf_set_horiz_scaling() : Obsolète : configure l'échelle horizontale du texte

Obsolète.

Voir [pdf_set_value](#) .

8.100.86 pdf_set_info_author() : Obsolète : remplit le champ d'auteur du document

Obsolète.

Voir [pdf_set_info](#) .

8.100.87 pdf_set_info_creator() : Obsolète : remplit le champ de créateur du document

Obsolète.

Voir [pdf_set_info](#) .

8.100.88 pdf_set_info_keywords() : Obsolète : remplit le champ de mots-clés du document

Obsolète.

Voir [pdf_set_info](#) .

8.100.89 pdf_set_info_subject() : Obsolète : remplit le champ de sujet du document

Obsolète.

Voir [pdf_set_info](#) .

8.100.90 pdf_set_info_title() : Obsolète : remplit le champ de titre du document

Obsolète.

Voir [pdf_set_info](#) .

8.100.91 pdf_set_info() : Remplit un champ de l'en-tête de document PDF

bool **pdf_set_info** (resource pdfdoc , string key , string value)

[pdf_set_info](#) remplit l'un des champs d'en-tête du document. Le paramètre key peut prendre l'une des valeurs suivantes : Subject , Title , Creator , Author , Keywords , ou une chaîne personnalisée. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.92 pdf_set_leading() : Obsolète : configure la distance entre deux lignes de texte

Obsolète.

Voir [pdf_set_value](#) .

8.100.93 pdf_set_parameter() : Modifie certains paramètres

bool **pdf_set_parameter** (resource pdfdoc , string key , string value)

[pdf_set_parameter](#) modifie les paramètres PDFlib de type texte. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [pdf_set_value](#) .

8.100.94 pdf_set_text_matrix() : Obsolète : configure la matrice de texte

Obsolète.

Voir [pdf_set_paramter](#) .

8.100.95 pdf_set_text_pos() : Modifie la position du texte

bool **pdf_set_text_pos** (resource pdfdoc , float x , float y)

[pdf_set_text_pos](#) modifie la position du texte en une nouvelle position définie par x et y . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.96 pdf_set_text_rendering() : Obsolète : détermine le rendu du texte

Obsolète.

Voir [pdf_set_value](#) .

8.100.97 pdf_set_text_rise() : Obsolète : configure l'élévation de texte

Obsolète.

Voir [pdf_set_value](#) .

8.100.98 pdf_set_value() : Modifie certains paramètres numériques

bool **pdf_set_value** (resource pdfdoc , string key , float value)

[pdf_set_value](#) modifie la valeur du paramètre de la PDFlib key par la valeur de value . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [pdf_set_parameter](#) .

8.100.99 pdf_set_word_spacing() : Obsolète : configure l'espace entre deux mots

Obsolète.

Voir [pdf_set_value](#) .

8.100.100 pdf_setcolor() : Configure la couleur de dessin et de remplissage

bool **pdf_setcolor** (resource pdfdoc , string type , string colorspace , float c1 , float c2 , float c3 , float c4)

[pdf_setcolor](#) configure la couleur de dessin et de remplissage. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre type peut être fill , stroke , ou both pour spécifier que la couleur est utilisée pour le remplissage, le dessin ou pour les deux. Le paramètre colorspace peut être gray , rgb , cmyk , spot ou pattern . Le paramètre c1 , c2 , c3 et c4 représente le composant de couleur pour l'espace de couleur colorspace . Excepté si précisé, le composant de couleur est un nombre à virgule flottante, de valeur comprise entre 0 et 1. Les paramètres c2 , c3 et c4 sont optionnels avant PHP 4.3.5 ou avec PDFlib inférieur à 5.

Pour les gris, seul c1 est utilisé.

Pour les couleurs rgb , les paramètres c1 , c2 et c3 spécifient le rouge, le vert et le bleu, respectivement.

Exemple avec [pdf_setcolor](#) : couleurs RGB

```
<?php
// Configure le blanc pour le dessin et le remplissage
pdf_setcolor($pdf, "both", "rgb", 1, 1, 1);
?>
```

Pour les couleurs cmyk , les paramètres c1 , c2 , c3 et c4 représentent le cyan, le magenta, le jaune et le noir, respectivement

Exemple avec [pdf_setcolor](#) : couleurs CMYK

```
<?php
// Configure le noir pour le dessin et le remplissage
pdf_setcolor($pdf, "both", "cmyk", 0, 0, 0, 1);
?>
```

Pour spot , c1 doit être une ressource de point de couleur, retournée par [pdf_makespotcolor](#) et c2 est une valeur de teinte, comprise entre 0 et 1 inclus.

Pour pattern , c1 doit être une ressource de pattern, retournée par [pdf_begin_pattern](#) .

8.100.101 pdf_setdash() : Configure le mode de pointillé

bool [pdf_setdash](#) (resource pdfdoc , float b , float w)

[pdf_setdash](#) configure le mode de pointillé à la taille de b pour les traits noirs, et w pour les traits blancs. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.102 pdf_setflat() : Configure la position à plat (flatness)

bool [pdf_setflat](#) (resource pdfdoc , float flatness)

[pdf_setflat](#) configure la position à plat, à une valeur comprise entre 0 et 100 inclus. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.103 pdf_setfont() : Configure la police courante

bool [pdf_setfont](#) (resource pdfdoc , int font , float size)

[pdf_setfont](#) configure la police courante avec la police font et la taille size . font est une ressource retournée par [pdf_findfont](#) . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [pdf_findfont](#) .

8.100.104 **pdf_setgray_fill()** : Configure la couleur de remplissage à un niveau de gris

bool **pdf_setgray_fill** (resource pdfdoc , float gray)

pdf_setgray_fill configure la couleur de remplissage à un niveau de gris, entre 0 et 1 inclus. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

PDFlib V4.0 : obsolète, utilisez pdf_setcolor .

8.100.105 **pdf_setgray_stroke()** : Configure la couleur de dessin à un niveau de gris

bool **pdf_setgray_stroke** (resource pdfdoc , float gray)

pdf_setgray_stroke configure la couleur de dessin à un niveau de gris, entre 0 et 1 inclus. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

PDFlib V4.0: obsolète, utilisez pdf_setcolor .
--

8.100.106 **pdf_setgray()** : Configure la couleur de dessin et de remplissage à un niveau de gris

bool **pdf_setgray** (resource pdfdoc , float gray)

pdf_setgray configure la couleur de dessin et de remplissage au niveau de gris gray . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

PDFlib V4.0 : obsolète , utilisez pdf_setcolor .
--

8.100.107 **pdf_setlinecap()** : Configure le paramètre de linecap

bool **pdf_setlinecap** (resource pdfdoc , int linecap)

pdf_setlinecap configure le paramètre linecap à une valeur entre 0 et 2 inclusivement.

8.100.108 **pdf_setlinejoin()** : Configure le paramètre de linejoin

bool **pdf_setlinejoin** (resource pdfdoc , int value)

pdf_setlinejoin configure le paramètre de linejoin à une valeur entre 0 et 2 inclus. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.109 pdf_setlinewidth() : Configure la largeur de ligne

bool **pdf_setlinewidth** (resource pdfdoc , float width)

pdf_setlinewidth configure la largeur de ligne à width . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.110 pdf_setmatrix() : Configure la matrice de transformation

bool **pdf_setmatrix** (resource pdfdoc , float a , float b , float c , float d , float e , float f)

pdf_setmatrix configure explicitement la matrice de transformation. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.111 pdf_setmiterlimit() : Configure la "miter limit"

bool **pdf_setmiterlimit** (resource pdfdoc , float miter)

pdf_setmiterlimit configure la "miter limit" à une valeur supérieure ou égale à 1. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.112 pdf_setpolydash() : Obsolète : configure des pointillés complexes

Obsolète.

Voir pdf_setdash .

8.100.113 pdf_setrgbcolor_fill() : Choisit la couleur utilisée pour le remplissage

bool **pdf_setrgbcolor_fill** (resource pdfdoc , float red_value , float green_value , float blue_value)

pdf_setrgbcolor_fill choisit la couleur utilisée pour le remplissage. Cette couleur est une couleur RGB. red_value représente le rouge, green_value le vert et blue_value le bleu. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note
PDFlib V4.0 : obsolète : utilisez <u>pdf_setcolor</u> .

8.100.114 pdf_setrgbcolor_stroke() : Choisit la couleur utilisée pour le dessin

bool **pdf_setrgbcolor_stroke** (resource pdfdoc , float red_value , float green_value , float blue_value)

`pdf_setrgbcolor_stroke` choisit la couleur utilisée pour le dessin. Cette couleur est une couleur RGB. `red_value` représente le rouge, `green_value` représente le vert et `blue_value` représente le bleu. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

PDFlib V4.0 : obsolète : utilisez <code>pdf_setcolor</code> .

8.100.115 `pdf_setrgbcolor()` : Choisit la couleur rgb de remplissage et de dessin

bool `pdf_setrgbcolor` (resource pdfdoc , float red_value , float green_value , float blue_value)

`pdf_setrgbcolor` choisit la couleur utilisée pour le dessin et le remplissage. Cette couleur est une couleur RGB. `red_value` représente le rouge, `green_value` représente le vert et `blue_value` représente le bleu. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

PDFlib V4.0 : obsolète : utilisez <code>pdf_setcolor</code> .

8.100.116 `pdf_show_boxed()` : Affiche le texte dans un cadre

int `pdf_show_boxed` (resource pdfdoc , string text , float left , float top , float width , float height , string mode , string feature)

`pdf_show_boxed` formate le texte `text` avec la police et la taille courante, pour qu'il soit contenu dans le rectangle de coin supérieur gauche (`left` , `top`), de largeur `width` et de hauteur `height` . Le mode de formatage peut être `left` , `right` , `center` , `justify` , ou `fulljustify` . Si la largeur `width` et la hauteur `height` sont nulles, une simple ligne, placée au point (`left` , `top`) dans le mode requis, sera dessinée. Le paramètre `feature` est optionnel avant PHP 4.3.5 ou avec PDFlib inférieur à 5.

`pdf_show_boxed` retourne le nombre de caractères qui n'ont pas pu être dessinés dans le rectangle défini. `pdf_show_boxed` retourne 0 si tous les caractères ont pu être dessinés, ou si `width` et `height` sont nuls.

8.100.117 `pdf_show_xy()` : Affiche un texte à une position donnée

bool `pdf_show_xy` (resource pdfdoc , string text , float x , float y)

`pdf_show_xy` affiche le texte `text` aux coordonnées `x` , `y` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.118 `pdf_show()` : Affiche le texte à la position courante

bool `pdf_show` (resource pdfdoc , string text)

`pdf_show` affiche le texte `text` à la position courante, avec la police et la taille courante.

8.100.119 pdf_skew() : Incline le système de coordonnées

bool **pdf_skew** (resource pdfdoc , float alpha , float beta)

pdf_skew incline le système de coordonnées de alpha degrés dans le sens des abscisses, et de beta degrés dans le sens des ordonnées. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.120 pdf_stringwidth() : Retourne la largeur d'un texte avec la police courante

float **pdf_stringwidth** (resource pdfdoc , string text , int font , float size)

pdf_stringwidth retourne la largeur du texte text en utilisant les paramètres font et size . Notez que font est une ressource de police retournée par la fonction pdf_findfont . Les paramètres font et size sont optionnels avant PHP 4.3.5 ou avec PDFlib inférieur à 5. Si aucun n'est spécifié, la police définie par la fonction pdf_setfont sera utilisée.

Note

Les paramètres font et size doivent être utilisés ensemble.

Voir aussi pdf_setfont et pdf_findfont .

8.100.121 pdf_stroke() : Dessine la ligne le long du chemin

bool **pdf_stroke** (resource pdfdoc)

pdf_stroke dessine la ligne le long du chemin courant, avec la couleur courante et la largeur de ligne courante, puis clôt le chemin. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.100.122 pdf_translate() : Effectue une translation de l'origine du système de coordonnées

bool **pdf_translate** (resource pdfdoc , float tx , float ty)

pdf_translate effectue une translation de l'origine du système de coordonnées d'un vecteur de (tx , ty).

8.101 Fonctions PDO

8.101.1 Introduction

L'extension PHP DATA OBJECTS (PDO) définit une excellente interface pour accéder à une base de données depuis PHP. Chaque pilote de base de données implémenté dans l'interface PDO peut utiliser des fonctionnalités spécifiques à chaque des bases de données en utilisant des extensions de fonctions. Notez que vous ne pouvez exécuter aucune fonction de base de données en utilisant l'extension PDO par elle-même ; vous devez utiliser un driver PDO spécifique à la base de données pour accéder au serveur de base de données.

PDO fournit une interface d'abstraction à l'accès de données, ce qui signifie que vous utilisez les mêmes fonctions pour exécuter des requêtes ou récupérer les données quelque soit la base de données utilisée. PDO ne fournit **pas** une abstraction de base de données : il ne réécrit pas le SQL, n'émule pas des fonctionnalités manquantes. Vous devriez utiliser une interface d'abstraction complète si vous avez besoin de cela.

PDO est fourni avec PHP 5.1 et est disponible en tant qu'extension PECL pour PHP 5.0 ; DO requiert les nouvelles fonctionnalités OO fournies par PHP 5 et donc, ne fonctionne pas avec les versions antérieures de PHP.

8.101.2 Installation

PHP 5.1 et supérieur sur les systèmes Unix

Si vous exécutez une version 5.1 de PHP, PDO est inclut dans cette distribution ; il devrait être automatiquement activé lorsque vous lancerez le script de configuration de PHP. Il est recommandé de compiler PDO en tant qu'extension partagée, ce qui vous permettra de mettre à jour PDO via PECL. Il est également recommandé d'activer le support zlib (pour l'installeur PEAR) lors de la compilation de PHP. Vous devriez également avoir besoin d'activer le driver PDO correspondant à la base de données de votre choix ; consultez la documentation pour les drivers PDO spécifiques à une base de données pour plus d'informations.

```
./configure --with-zlib --enable-pdo=shared
```

Après avoir installé PDO en tant que module partagé, vous devez éditer votre fichier php.ini afin que l'extension PDO soit chargé automatiquement lorsque PHP est lancé. Vous devez y activer également le driver spécifique à votre base de données ; assurez-vous qu'ils sont listés après la ligne pdo.so, car PDO doit être initialisé avant le chargement des extensions spécifiques à la base de données. Si vous avez compiler les extensions statiquement, vous pouvez sauter cette étape.
extension=pdo.so

Le fait d'avoir PDO en tant que module partagé vous permet d'exécuter la commande pear upgrade pdo pour mettre à jour votre version de PDO sans pour autant avoir besoin de recompiler complètement PHP. Notez que si vous faites cela, vous devez également mettre à jour votre driver PDO spécifique à votre base de données dans le même temps.

PHP 5.0 et supérieur sur les systèmes Unix

PDO est disponible en tant qu'extension PECL depuis <http://pecl.php.net/package/pdo> .

L'installation peut être faite via la commande pear ; ceci est activé par défaut lorsque vous configurez PHP. Vous devriez vous assurer que PHP a été configuré avec --with-zlib pour pouvoir

utiliser la commande pear afin de gérer les paquets compressés.

Exécutez la commande suivante pour télécharger, compiler et installer la dernière version stable de PDO :

```
pear install pdo
```

Si PDO est toujours en version BETA (et au moment de l'écriture de ces lignes, il l'est), vous devez demander à l'utilitaire PEAR de récupérer le paquet BETA. Au lieu d'exécuter la commande ci-dessus, lancez la commande suivante :

```
pear install pdo-beta
```

La commande pear installe automatiquement le module PDO dans votre dossier d'extensions PHP. Pour activer l'extension PDO sous les systèmes Linux ou Unix, vous devez ajouter la ligne suivante dans votre fichier php.ini :

```
extension=pdo.so
```

Pour plus d'informations sur la compilation de paquets PECL, consultez la section sur l' [installation de PECL](#) du manuel.

Utilisateurs de Windows avec PHP 5.1 et supérieur

PDO ainsi que tous les drivers principaux interagissent avec PHP en tant qu'extensions partagées, et ont tout simplement besoin d'être activés en édiant le fichier php.ini :

```
extension=php_pdo.dll
```

Ensuite, choisissez les autres fichiers DLL spécifiques à votre base de données et utilisez soit la fonction `dl` pour les charger au moment de l'exécution ou activer-les dans le fichier php.ini en dessous de la ligne `php_pdo.dll` . Par exemple :

```
extension=php_pdo.dll
extension=php_pdo_firebird.dll
extension=php_pdo_mssql.dll
extension=php_pdo_mysql.dll
extension=php_pdo_oci.dll
extension=php_pdo_oci8.dll
extension=php_pdo_odbc.dll
extension=php_pdo_pgsqll.dll
extension=php_pdo_sqlite.dll
```

Ces bibliothèques DDLs doivent exister dans le dossier système `extension_dir` .

8.101.3 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

Nom	Par défaut	Modifiable	Historique
pdo.dsn.*		php.ini seulement	

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

```
pdo.dsn.* string
```

Définit un alias DSN. Voyez [fonction.PDO-construct](#) pour une explication complète.

8.101.4 Drivers PDO

Les drivers suivants sont actuellement implémentés dans l'interface PDO :

Nom du driver	Bases de données supportées
<u>PDO_DBLIB</u>	FreeTDS / Microsoft SQL Server / Sybase
<u>PDO_FIREBIRD</u>	Firebird/Interbase 6
<u>PDO_MYSQL</u>	MySQL 3.x/4.0
<u>PDO_OCI</u>	Oracle Call Interface
<u>PDO_ODBC</u>	ODBC v3 (IBM DB2 unixODBC et win32 ODBC)
<u>PDO_PGSQL</u>	PostgreSQL
<u>PDO_SQLITE</u>	SQLite 3 et SQLite 2

8.101.5 Connexions et gestionnaire de connexion

Les connexions sont établies en créant des instances de la classe de base de PDO. Peut importe quel driver vous voulez utiliser ; vous utilisez toujours le nom de la classe PDO. Le constructeur accepte des paramètres pour spécifier la source de la base de données (connue en tant que DSN) et optionnellement, le nom d'utilisateur et le mot de passe (s'il y en a un).

Connexion à MySQL
<pre><?php \$dbh = new PDO('mysql:host=localhost;dbname=test', \$user, \$pass); ?></pre>

S'il y a des erreurs de connexion, un objet PDOException est lancé. Vous pouvez attrapper cette exception si vous voulez gérer cette erreur, ou laisser le gestionnaire globale d'exception défini via la fonction `set_exception_handler` la traiter.

Gestion des erreurs de connexion
<pre><?php try { \$dbh = new PDO('mysql:host=localhost;dbname=test', \$user, \$pass); foreach (\$dbh->query('SELECT * from FOO') as \$row) { print_r(\$row); } \$dbh = null; } catch (PDOException \$e) { print "Erreur ! : " . \$e->getMessage() . "
"; die(); } ?></pre>

Attention
<p>Si votre application n'attrape pas les exceptions lancées depuis le constructeur PDO, l'action par défaut du moteur zend est de terminer le script et d'afficher une trace. Cette trace devrait révéler des détails complets sur la connexion à la base de données, incluant le nom d'utilisateur et le mot de passe. Il en est donc de votre responsabilité d'attrapper cette exception, soit explicitement (via l'instruction <code>catch</code>) ou implicitement via la fonction <code>set_exception_handler</code>.</p>

Lorsque la connexion à la base de données a réussi, une instance de la classe PDO est retourné à votre script. La connexion est active tant que l'objet PDO l'est. Pour clôturer la connexion, vous devez détruire l'objet en vous assurant que toutes ses références sont effacées--vous pouvez faire cela en assignant NULL à la variable gérant l'objet. Si vous ne le faites pas explicitement, PHP fermera automatiquement la connexion lorsque le script arrivera à la fin.

Fermeture d'un connexion

```
<?php
$dbh = new PDO('mysql:host=localhost;dbname=test', $user, $pass);
// utiliser la connexion ici

// et maintenant, fermez-la !
$dbh = null;
?>
```

Beaucoup d'applications web utilisent des connexions persistantes aux serveurs de base de données. Les connexions persistantes ne sont pas fermées à la fin du script, mais sont mises en cache et réutilisées lorsqu'un autre script demande une connexion en utilisant les mêmes paramètres. Le cache des connexions persistantes vous permet d'éviter d'établir une nouvelle connexion à chaque fois qu'un script doit accéder à une base de données, rendant l'application web plus rapide.

Connexions persistantes

```
<?php
$dbh = new PDO('mysql:host=localhost;dbname=test', $user, $pass, array(
    PDO::ATTR_PERSISTENT => true
));
?>
```

Note

Si vous utilisez le driver PDO ODBC et que votre bibliothèque ODBC supporte le pool de connexion ODBC (unixODBC et Windows le supportent tous les deux ; peut être plus), alors, il est recommandé de ne pas utiliser les connexions persistantes PDO mais plutôt laisser le pool de connexion ODBC mettre en cache les connexions. Le pool de connexion ODBC est partagé avec les autres modules dans le processus ; si PDO met en cache la connexion, alors, cette connexion ne sera jamais retournée par le pool de connexion ODBC, faisant que plusieurs connexions sont créées pour les autres modules.

8.101.6 Transactions et validation automatique (auto-commit)

Maintenant que vous êtes connecté via PDO, vous devez comprendre comme PDO gère les transactions avant d'exécuter des requêtes. Si vous n'avez jamais utilisé les transactions, elles offrent 4 fonctionnalités majeures : Automatisation, Consistance, Isolation et Durabilité (ACID). En d'autres termes, n'importe quel travail mené à bien dans une transaction, même s'il est effectué par étapes, est garanti pour être appliqué à la base de données sans risque, et sans interférence pour les autres connexions, quand il est validé. Le travail des transactions peut également être automatiquement annulé à votre demande (en supposant que vous n'avez encore rien validé), ce qui rend la gestion des erreurs bien plus simple dans vos scripts.

Les transactions sont typiquement implémentées pour appliquer toutes vos modifications en une seule fois ; ceci a le bel effet d'éprouver drastiquement l'efficacité de vos mises à jour. Dans d'autres termes, les transactions rendent vos scripts plus rapides et potentiellement plus robuste (vous devez les utiliser correctement pour avoir ces bénéfices).

Malheureusement, toutes les bases de données ne supportent pas les transactions, donc, PDO doit s'exécuter en mode "auto-commit" lorsque vous ouvrez pour la première fois la connexion. Le mode "auto-commit" signifie que toutes les requêtes que vous exécutez ont leurs transactions implicites, si la base de données le supporte ou aucune transaction si la base de données ne les supporte pas. Si vous avez besoin d'une transaction, vous devez utiliser la méthode [PDO::beginTransaction](#) pour l'initialiser. Si le driver utilisé ne supporte pas les transactions, une exception PDO sera lancé (en accord avec votre gestionnaire d'erreurs : ceci est toujours une erreur sérieuse). Une fois que

vous êtes dans une transaction, vous devez utiliser la fonction `PDO::commit` ou la fonction `PDO::rollBack` pour la terminer, suivant le succès de votre code durant la transaction.

Lorsque le script se termine ou lorsque la connexion est sur le point de se fermer, si vous avez une transaction en cours, PDO l'annulera automatiquement. Ceci est une mesure de sécurité afin de garantir la consistance de vos données dans le cas où le script se termine d'une façon inattendue--si vous ne validez pas explicitement la transaction, alors, on présume que quelque chose s'est mal passé et l'annulation de la transaction intervient afin de garantir la sécurité de vos données.

Attention

L'annulation automatique intervient si vous avez initialisé la transaction via `PDO::beginTransaction`. Si vous avez manuellement exécuté une requête qui commence une transaction, PDO n'a aucun moyen de le savoir et donc, n'annulera pas automatiquement cette transaction si quelque chose s'est mal passé.

Exécution d'un groupe dans une transaction

Dans l'exemple suivant, supposons que nous allons créer un jeu d'entrées pour un nouvel employé, dont le numéro d'ID sera 23. En plus des données basiques sur cette personne, vous devez également lui enregistrer son salaire. Il est très simple d'effectuer deux mises à jour séparées, mais en les enfermant dans les appels des fonctions `PDO::beginTransaction` et `PDO::commit`, nous garantissons que personne ne pourra voir ces modifications tant qu'elles ne seront pas complètes. Si quelque chose tourne mal, le block de capture annulera toutes les modifications effectuées depuis le début de la transaction et affichera un message d'erreur.

```
<?php
try {
    $dbh = new PDO('odbc:SAMPLE', 'db2inst1', 'ibmdb2',
        array(PDO::ATTR_PERSISTENT => true));
    echo "Connecté\n";
    $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $dbh->beginTransaction();
    $dbh->exec("insert into staff (id, first, last) values (23, 'Joe', 'Bloggs')");
    $dbh->exec("insert into salarychange (id, amount, changedate)
values (23, 50000, NOW())");
    $dbh->commit();

} catch (Exception $e) {
    $dbh->rollBack();
    echo "Échec : " . $e->getMessage();
}
?>
```

Vous n'êtes pas limité dans le nombre de mises à jour dans une transaction ; vous pouvez également y effectuer des requêtes complexes et bien sûr, utiliser ces informations pour construire d'autres mises à jour et requêtes ; durant l'activité de la transaction, vous êtes garantie que personne d'autre ne peut effectuer des modifications alors que vous êtes au milieu de vos modifications. En réalité, ceci n'est pas 100% correct, mais cela reste une très bonne introduction aux transactions.

8.101.7 Requêtes préparées et procédures stockées

La plupart des bases de données supporte le concepte des requêtes préparées. Qu'est-ce donc ? Vous pouvez les voir comme une sorte de template compilée pour le SQL que vous voulez exécuter, qui peut être personnalisé en utilisant des variables en guise de paramètres. Les requêtes préparées offrent deux fonctionnalités essentielles :

- La requête ne doit être analysée (ou préparée) qu'une seule fois, mais peut être exécutée plusieurs fois avec des paramètres identiques ou différents. Lorsque la requête est préparée, la base de données va analyser, compiler et optimiser son plan pour exécuter la requête. Pour les requêtes complexes, ce processus peut prendre assez de temps, ce qui peut ralentir vos applications si vous devez répéter la même requête plusieurs fois avec différents paramètres. En utilisant les requêtes préparées, vous évitez ainsi de répéter le cycle analyser/compilation/optimisation. Pour résumer, les requêtes préparées utilisent moins de ressource et s'exécutent plus rapidement.
- Les paramètres pour préparer les requêtes n'ont pas besoin d'être entre guillemets ; le driver le gère pour vous. Si votre application utilise exclusivement les requêtes préparées, vous pouvez être sûr qu'aucune injection SQL n'est possible (Cependant, si vous construisez d'autres parties de la requête en vous basant sur des entrées utilisateurs, vous continuez à prendre un risque).

Les requêtes préparées sont tellement pratiques que c'est l'unique fonctionnalité que PDO émule pour les drivers qui ne les supportent pas. Ceci assure de pouvoir utiliser la même technique pour accéder aux données, sans se soucier des capacités de la base de données.

Insertions répétitives en utilisant les requêtes préparées

Cet exemple effectue une requête INSERT en y substituant un nom et une valeur pour les marqueurs nommés.

```
<?php
$stmt = $dbh->prepare("INSERT INTO REGISTRY (nom, valeur) VALUES (:nom, :valeur)");
$stmt->bindParam(':nom', $nom);
$stmt->bindParam(':valeur', $valeur);

// insertion d'une ligne
$nom = 'one';
$valeur = 1;
$stmt->execute();

// insertion d'une autre ligne avec des valeurs différentes
$nom = 'two';
$valeur = 2;
$stmt->execute();
?>
```

Insertions répétées en utilisant des requêtes réparées

Cet exemple effectue une requête INSERT en y substituant un nom et une valeur pour les marqueurs ? .

```
<?php
$stmt = $dbh->prepare("INSERT INTO REGISTRY (nom, valeur) VALUES (?, ?)");
$stmt->bindParam(1, $nom);
$stmt->bindParam(2, $valeur);

// insertion d'une ligne
$nom = 'one';
$valeur = 1;
$stmt->execute();

// insertion d'une autre ligne avec différentes valeurs
$nom = 'two';
$valeur = 2;
$stmt->execute();
?>
```

Récupération des données en utilisant des requêtes préparées

Cet exemple récupère des données basées sur la valeur d'une clé fournie par un formulaire. L'entrée utilisateur est automatiquement échappée, il n'y a donc aucun risque d'attaque par injection SQL.

```
<?php
$stmt = $dbh->prepare("SELECT * FROM REGISTRY where name = ?");
if ($stmt->execute(array($_GET['name']))) {
    while ($row = $stmt->fetch()) {
        print_r($row);
    }
}
?>
```

Si le driver de la base de données le supporte, vous pouvez également lier des paramètres aussi bien pour l'entrée que pour la sortie. Les paramètres de sortie sont utilisés typiquement pour récupérer les valeurs d'une procédure stockée. Les paramètres de sortie sont un peu plus complexe à utiliser que les paramètres d'entrée car vous devez savoir la longueur d'un paramètre donné pour atteindre lorsque vous le liez. Si la valeur retournée est plus longue que la taille qui vous auriez suggéré, une erreur sera émise.

Appel d'une procédure stockée avec un paramètre de sortie

```
<?php
$stmt = $dbh->prepare("CALL sp_returns_string(?)");
$stmt->bindParam(1, $return_value, PDO::PARAM_STR, 4000);

// Appel de la procédure stockée
$stmt->execute();

print "La procédure a retourné : $return_value\n";
?>
```

Vous devez également spécifier les paramètres qui gèrent les valeurs aussi bien pour l'entrée que pour la sortie ; la syntaxe est similaire aux paramètres de sortie. Dans le prochain exemple, la chaîne 'Bonjour' est passée à la procédure stockée et lorsqu'elle retourne la valeur, 'Bonjour' est remplacé par la valeur retournée par la procédure.

Appel d'une procédure stockée avec un paramètre d'entrée/sortie

```
<?php
$stmt = $dbh->prepare("CALL sp_takes_string_returns_string(?)");
$value = 'hello';
$stmt->bindParam(1, $value, PDO::PARAM_STR|PDO::PARAM_INPUT_OUTPUT, 4000);

// appel de la procédure stockée
$stmt->execute();

print "La procédure a retourné : $value\n";
?>
```

8.101.8 Erreurs et gestionnaire des erreurs

PDO offre la possibilité de gérer les erreurs de 3 façons différentes, pour coller au mieux au style de votre application.

- PDO::ERRMODE_SILENT

C'est le mode par défaut. PDO définit simplement le code erreur pour vous afin de

l'inspecter en utilisant les méthodes `PDO::errorCode` et `PDO::errorInfo` sur la requête ou sur les objets de la base de données ; si l'erreur est issue d'un appel à un objet de requête, vous devriez invoquer les méthodes `PDOStatement::errorCode` ou `PDOStatement::errorInfo` sur cet objet. Si l'erreur est issue d'un appel sur un objet de la base de données, vous devriez invoquer à la place les méthodes de cet objet.

- `PDO::ERRMODE_WARNING`

En plus de définir le code erreur, PDO émettra un message de niveau `E_WARNING`. Ce comportement est très utile lors de la phase de test/débogage, si vous voulez juste voir le problème qui survient sans pour autant stopper le flot de l'application.

- `PDO::ERRMODE_EXCEPTION`

En plus de définir le code erreur, PDO lancera une exception `PDOException` et y définit ces propriétés pour refléter le code erreur et l'information sur l'erreur. Ce comportement est utile également durant la période de débogage car il va stopper l'application à l'endroit même de l'erreur et pointer l'endroit où se trouve potentiellement l'erreur dans votre code (souvenez-vous : les transactions sont automatiquement annulées si l'exception fait que le script se termine).

Le mode lançant des exceptions est également utile pour structurer la gestion de vos erreurs plus clairement que par la traditionnelle gestion des erreurs de PHP et avec moins de code, plutôt que le mode silencieux et l'analyse explicite des valeurs retournées par chaque appel à la base de données.

Lisez la documentation sur les [exceptions](#) pour plus de détails sur les exceptions en PHP.

PDO standardise les codes erreurs en utilisant SQL-92 SQLSTATE ; les drivers individuels PDO sont responsables du mapping de leurs propres codes avec les codes SQLSTATE appropriées. La méthode `PDO::errorCode` retourne un seul code SQLSTATE. Si vous avez besoin de plus d'informations sur une erreur, PDO fournit également la méthode `PDO::errorInfo` qui retourne un tableau contenant le code SQLSTATE, le code erreur spécifique du driver et la chaîne erreur spécifique au driver.

8.101.9 Les objets larges (LOBs)

À un moment dans votre application, vous pourriez avoir besoin de stocker de larges données dans votre base de données. "Large" signifie typiquement des données d'environ 4ko ou plus, bien que certaines bases de données peuvent gérer plus de 32ko avant que les données deviennent "larges". Les objets larges peuvent être de nature textuelle ou binaire. PDO vous permet de travailler avec ce type de larges données en utilisant le code type `PDO::PARAM_LOB` dans vos appels aux fonctions `PDOStatement::bindParam` ou `PDOStatement::bindColumn` . `PDO::PARAM_LOB` demande à PDO de mapper les données en un flux que vous pourrez manipuler en utilisant l' [API PHP sur les flux](#) .

Affichage d'une image depuis une base de données

Cet exemple lie un LOB dans une variable nommée `$lob` et l'envoie au navigateur en utilisant la fonction `fpassthru` . Vû qu'un LOB est représenté en un flux, les fonctions comme `fgets` , `fread` et `stream_get_contents` peuvent être utilisées sur ce flux.

```
<?php
$db = new PDO('odbc:SAMPLE', 'db2inst1', 'ibmdb2');
$stmt = $db->prepare("select contenttype, imagedata from images where id=?");
$stmt->execute(array($_GET['id']));
$stmt->bindColumn(1, $type, PDO::PARAM_STR, 256);
```

```
$stmt->bindColumn(2, $lob, PDO::PARAM_LOB);
$stmt->fetch(PDO::FETCH_BOUND);

header("Content-Type: $type");
fpassthru($lob);
?>
```

Insertion d'une image dans une base de données

Cet exemple ouvre un fichier et passe le pointeur de fichier à PDO pour l'insérer en tant que LOB. PDO fera son possible pour récupérer le contenu du fichier et l'insérer dans la base de données de la manière la plus efficace possible.

```
<?php
$db = new PDO('odbc:SAMPLE', 'db2inst1', 'ibmdb2');
$stmt = $db->prepare("insert into images (id, contenttype, imagedata) values (?, ?, ?)");
$id = get_new_id(); // fonction pour allouer un nouvel ID

// assumons que nous récupérons un fichier depuis un formulaire
// vous pouvez trouver plus de détails dans la documentation de PHP

$fp = fopen($_FILES['file']['tmp_name'], 'rb');

$stmt->bindParam(1, $id);
$stmt->bindParam(2, $_FILES['file']['type']);
$stmt->bindParam(3, $fp, PDO::PARAM_LOB);

$stmt->beginTransaction();
$stmt->execute();
$stmt->commit();
?>
```

Insertion d'une image dans une base de données Oracle

Oracle requiert une syntaxe légèrement différente pour y insérer un LOB depuis un fichier. Il est également essentielle d'effectuer votre insertion au sein d'une transaction, sinon, votre nouveau LOB sera inséré avec une longueur de zéro :

```
<?php
$db = new PDO('oci:', 'scott', 'tiger');
$stmt = $db->prepare("insert into images (id, contenttype, imagedata) " .
"VALUES (?, ?, EMPTY_BLOB()) RETURNING imagedata INTO ?");
$id = get_new_id(); // fonction pour allouer un nouvel ID

// assumons que nous récupérons un fichier depuis un formulaire
// vous pouvez trouver plus de détails dans la documentation de PHP

$fp = fopen($_FILES['file']['tmp_name'], 'rb');

$stmt->bindParam(1, $id);
$stmt->bindParam(2, $_FILES['file']['type']);
$stmt->bindParam(3, $fp, PDO::PARAM_LOB);

$stmt->beginTransaction();
$stmt->execute();
$stmt->commit();
?>
```

8.101.10 Classes pré-définies

8.101.10.1 PDO

Représente une connexion entre PHP et un serveur de base de données.

8.101.10.1.1 Constructeur

- PDO - construit un nouvel objet PDO

8.101.10.1.2 Méthodes

- beginTransaction - commence une transaction
- commit - valide une transaction
- errorCode - récupère un code erreur, s'il y en a, depuis la base de données
- errorInfo - récupère un tableau contenant les informations sur l'erreur, s'il y en a, depuis la base de données
- exec - exécute une requête SQL et retourne le nombre de lignes affectées
- getAttribute - récupère un attribut d'une connexion à une base de données
- lastInsertId - récupère la valeur de la dernière ligne insérée dans une table
- prepare - prépare une requête SQL pour exécution
- query - exécute une requête SQL et retourne le jeu de résultats
- quote - retourne une version quotée d'une chaîne pour utilisation dans une requête SQL
- rollBack - annule une transaction
- setAttribute - définit un attribut d'une connexion à une base de données

8.101.10.2 PDOStatement

Représente une requête préparée et, après, exécution de la requête, un jeu de résultat associatif.

8.101.10.2.1 Méthodes

- bindColumn - lie une variable PHP à une colonne de sortie dans un jeu de résultats
- bindParam - lie une variable PHP à un paramètre dans une requête préparée
- bindValue - lie une valeur à un paramètre dans une requête préparée
- closeCursor - ferme le curseur, autorisant la requête à être exécutée une nouvelle fois
- columnCount - retourne le nombre de colonnes dans un jeu de résultats
- errorCode - récupère le code erreur, s'il y en a, depuis une requête
- errorInfo - récupère un tableau contenant les informations sur une erreur, s'il y en a, depuis une requête
- execute - exécute une requête préparée
- fetch - récupère une ligne depuis un jeu de résultats
- fetchAll - récupère un tableau contenant tous les lignes depuis un jeu de résultats
- fetchColumn - retourne les données depuis une colonne d'un jeu de résultats
- getAttribute - récupère un attribut PDOStatement
- getColumnMeta - récupère les méta-informations pour une colonne d'un jeu de résultats
- nextRowset - récupère la ligne suivante (du jeu de résultats)
- rowCount - retourne le nombre de lignes affectées par l'exécution d'une requête SQL
- setAttribute - définit un attribut PDOStatement
- setFetchMode - définit le mode de récupération pour une PDOStatement

8.101.10.3 PDOException

Représente une erreur émise par PDO. Vous ne devriez pas lancer une exception PDOException depuis votre propre code. Lisez la documentation sur les [Exceptions](#) pour plus de détails sur les exceptions en PHP.

La classe PDOException

```
<?php
class PDOException extends Exception
{
public $errorInfo = null;    // correspond à PDO::errorInfo()
// ou PDOStatement::errorInfo()
protected $message;        // message d'erreur textuel
// utiliser Exception::getMessage() pour y accéder
protected $code;           // code erreur SQLSTATE
// utiliser Exception::getCode() pour y accéder
}
?>
```

8.101.11 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Attention

PDO utilise les constantes de classe depuis PHP 5.1. Les versions antérieures utilisent les constantes globales sous la forme PDO_PARAM_BOOL .

PDO::PARAM_BOOL ([entier](#))

Représente le type de données booléen.

PDO::PARAM_NULL ([entier](#))

Représente le type de données NULL SQL.

PDO::PARAM_INT ([entier](#))

Représente le type de données INTEGER SQL.

PDO::PARAM_STR ([entier](#))

Représente les types de données CHAR, VARCHAR ou les autres types de données sous forme de chaîne de caractères SQL.

PDO::PARAM_LOB ([entier](#))

Représente le type de données "objet large" SQL.

PDO::PARAM_STMT ([entier](#))

Représente un type de jeu de résultats. N'est actuellement pas supporté par tous les drivers.

PDO::PARAM_INPUT_OUTPUT ([entier](#))

Spécifie que le paramètre est un paramètre INOUT pour une procédure stockée. Vous devez utiliser l'opérateur OR avec un type de données explicite PDO::PARAM_*.

PDO::FETCH_LAZY ([entier](#))

Spécifie que la méthode de récupération doit retourner chaque ligne en tant qu'objet avec les noms de variables correspondant aux noms des colonnes retournées dans le jeu de résultats. PDO::FETCH_LAZY crée les noms des variables de l'objet comme ils sont rencontrés.

PDO::FETCH_ASSOC ([entier](#))

Spécifie que la méthode de récupération doit retourner chaque ligne dans un tableau indexé par les noms des colonnes comme elles sont retournées dans le jeu de résultats correspondant. Si le jeu de résultats contient de multiples colonnes avec le

- même nom, PDO::FETCH_ASSOC retourne une seule valeur par nom de colonne.
- PDO::FETCH_NAMED (entier)
Spécifie que la méthode de récupération doit retourner chaque ligne dans un tableau indexé par les noms des colonnes comme elles sont retournées dans le jeu de résultats correspondant. Si le jeu de résultats contient de multiples colonnes avec le même nom, PDO::FETCH_NAMED retourne un tableau de valeurs par nom de colonne.
- PDO::FETCH_NUM (entier)
Spécifie que la méthode de récupération doit retourner chaque ligne dans un tableau indexé par le numéro des colonnes comme elles sont retournées dans le jeu de résultats correspondant, en commençant à 0.
- PDO::FETCH_BOTH (entier)
Spécifie que la méthode de récupération doit retourner chaque ligne dans un tableau indexé par les noms des colonnes ainsi que leurs numéros, comme elles sont retournées dans le jeu de résultats correspondant, en commençant à 0.
- PDO::FETCH_OBJ (entier)
Spécifie que la méthode de récupération doit retourner chaque ligne dans un objet avec les noms de propriétés correspondant aux noms des colonnes comme elles sont retournées dans le jeu de résultats.
- PDO::FETCH_BOUND (entier)
Spécifie que la méthode de récupération doit retourner TRUE et assigner les valeurs des colonnes du jeu de résultats dans les variables PHP auxquelles elles sont liées avec la méthode `PDOStatement::bindParam()` ou la méthode `PDOStatement::bindColumn` .
- PDO::FETCH_COLUMN (entier)
Spécifie que la méthode de récupération doit retourner uniquement une seule colonne demandée depuis la prochaine ligne du jeu de résultats.
- PDO::FETCH_CLASS (entier)
Spécifie que la méthode de récupération doit retourner une nouvelle instance de la classe demandée, liant les colonnes aux propriétés nommées dans la classe.
- PDO::FETCH_INTO (entier)
Spécifie que la méthode de récupération doit mettre à jour une instance existante de la classe demandée, liant les colonnes aux propriétés nommées dans la classe.
- PDO::FETCH_FUNC (entier)
- PDO::FETCH_GROUP (entier)
- PDO::FETCH_UNIQUE (entier)
- PDO_ATTR_AUTOCOMMIT (entier)
Si la valeur vaut FALSE , PDO tente de désactiver l'auto-validation lorsque la connexion commence une transaction.
- PDO::ATTR_PREFETCH (entier)
Définir la taille de la pré-récupération vous permet d'accroître les performances de votre application. Toutes les combinaisons bases de données / drivers ne supportent pas cette fonctionnalité. Ceci accroît les performances au détriment de la consommation de mémoire vive.
- PDO::ATTR_TIMEOUT (entier)
Définit la valeur d'attente en secondes pour les communications avec la base de données.
- PDO::ATTR_ERRMODE (entier)
Voir la section sur les erreurs et la gestion des erreurs pour plus d'informations sur cet attribut.
- PDO::ATTR_SERVER_VERSION (entier)
Attribut en lecture seul ; il retourne des informations sur la version de la base de données à laquelle PDO est connecté.
- PDO::ATTR_CLIENT_VERSION (entier)

Attribut en lecture seul ; il retourne des informations sur la version de la bibliothèque cliente utilisée par PDO.

PDO::ATTR_SERVER_INFO ([entier](#))

Attribut en lecture seul ; il retourne quelques meta-informations sur le serveur de base de données auquel PDO est connecté.

PDO::ATTR_CONNECTION_STATUS ([entier](#))

PDO::ATTR_CASE ([entier](#))

Force les noms des colonnes dans une casse spécifiée par les constantes PDO::CASE_*.

PDO::ATTR_CURSOR_NAME ([entier](#))

Récupère ou définit le nom à utiliser pour un curseur. Très utile lors de l'utilisation de curseurs scrollables et des mises à jour positionnées.

PDO::ATTR_CURSOR ([entier](#))

Sélectionne le type de curseur. PDO supporte actuellement soit PDO::CURSOR_FWDONLY , soit PDO::CURSOR_SCROLL . Conserver PDO::CURSOR_FWDONLY tant que vous savez que vous avez besoin d'un curseur scrollable.

PDO::ATTR_DRIVER_NAME ([chaîne de caractères](#))

Retourne le nom du driver.

Utilisation de PDO::ATTR_DRIVER_NAME
<pre><?php if (\$db->getAttribute(PDO::ATTR_DRIVER_NAME) == 'mysql') { echo "Utilisation de mysql ; fait quelque chose de spécifique à mysql ici\n"; } ?></pre>

PDO::ATTR_ORACLE_NULLS ([entier](#))

Convertit les chaînes vides en valeurs NULL SQL dans les données récupérées.

PDO::ATTR_PERSISTENT ([entier](#))

Demande une connexion persistante, plutôt que de créer une nouvelle connexion. Voir les [connexions](#) et le [gestionnaire de connexion](#) pour plus d'informations sur cet attribut.

PDO::ATTR_FETCH_CATALOG_NAMES (integer)

Ajoute le contenu du catalogue de noms dans chaque nom de colonnes retourné dans le jeu de résultat. Le catalogue de noms et les noms de colonnes sont séparés par un point (.). Le support de cet attribut n'est pas disponible pour tous les drivers ; il peut ne pas être disponible pour votre driver.

PDO::ATTR_FETCH_TABLE_NAMES (integer)

Ajoute le contenu de la table de noms dans chaque nom de colonne retourné dans le jeu de résultats. La table de nom et les noms de colonnes sont séparés par un point (.). Le support de cet attribut n'est pas disponible pour tous les drivers ; il peut ne pas être disponible pour votre driver.

PDO::ERRMODE_SILENT ([entier](#))

N'envoie pas d'erreur ni d'exception si une erreur survient. Le développeur doit explicitement vérifier les erreurs. C'est le mode par défaut. Voir les [erreurs et la gestion des erreurs](#) pour plus d'informations sur cet attribut.

PDO::ERRMODE_WARNING ([entier](#))

Envoie une erreur de niveau E_WARNING si une erreur survient. Voir les [erreurs et la gestion des erreurs](#) pour plus d'informations sur cet attribut.

PDO::ERRMODE_EXCEPTION ([entier](#))

Lance une exception PDOException si une erreur survient. Voir les [erreurs et la gestion des erreurs](#) pour plus d'informations sur cet attribut.

PDO::CASE_NATURAL ([entier](#))

Laisse les noms de colonnes comme retournés par le driver de base de données.

`PDO::CASE_LOWER` ([entier](#))

Force les noms de colonnes en minuscule.

`PDO::CASE_UPPER` ([entier](#))

Force les noms des colonnes en majuscule.

`PDO::FETCH_ORI_NEXT` ([entier](#))

Récupère la prochaine ligne d'un jeu de résultats. Valide seulement pour les curseurs scrollables.

`PDO::FETCH_ORI_PRIOR` ([entier](#))

Récupère la ligne précédente d'un jeu de résultats. Valide seulement pour les curseurs scrollables.

`PDO::FETCH_ORI_FIRST` ([entier](#))

Récupère la première ligne d'un jeu de résultats. Valide seulement pour les curseurs scrollables.

`PDO::FETCH_ORI_LAST` ([entier](#))

Récupère la dernière ligne d'un jeu de résultats. Valide seulement pour les curseurs scrollables.

`PDO::FETCH_ORI_ABS` ([entier](#))

Récupère la ligne demandée par un numéro de ligne d'un jeu de résultats. Valide seulement pour les curseurs scrollables.

`PDO::FETCH_ORI_REL` ([entier](#))

Récupère la ligne demandé par une position relative à la position courante du curseur d'un jeu de résultats. Valide seulement pour les curseurs scrollables.

`PDO::CURSOR_FWDONLY` ([entier](#))

Crée un objet PDOStatement avec un curseur uniquement de retour. C'est le choix par défaut pour le curseur, car il est rapide et l'accès aux données est commune pour les masques en PHP.

`PDO::CURSOR_SCROLL` ([entier](#))

Crée un objet PDOStatement avec un curseur scrollable. Passez la constante `PDO::FETCH_ORI_*` pour contrôler les lignes récupérées du jeu de résultats.

`PDO::ERR_NONE` ([chaîne de caractères](#))

Correspond à `SQLSTATE '00000'`, ce qui signifie que la requête SQL a réussi sans erreur, ni avertissement. Cette constante est utile lorsque vous utilisez `PDO::errorCode` ou `PDOStatement::errorCode` pour déterminer si une erreur est survenue. Cependant, vous devez déjà savoir si c'est le cas en examinant le code retourné par la méthode qui a lancée l'erreur.

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Drivers PDO](#)
- [Connexions et gestionnaire de connexion](#)
- [Transactions et validation automatique \(auto-commit\)](#)
- [Requêtes préparées et procédures stockées](#)
- [Erreurs et gestionnaire des erreurs](#)
- [Les objets larges \(LOBs\)](#)
- [Classes pré-définies](#)
- [Constantes pré-définies](#)
- [PDO::beginTransaction](#)
- [PDO::commit](#)
- [PDO::__construct](#)
- [PDO::errorCode](#)
- [PDO::errorInfo](#)
- [PDO::exec](#)
- [PDO::getAttribute](#)

- [PDO::getAvailableDrivers](#)
- [PDO::lastInsertId](#)
- [PDO::prepare](#)
- [PDO::query](#)
- [PDO::quote](#)
- [PDO::rollBack](#)
- [PDO::setAttribute](#)
- [PDOStatement::bindColumn](#)
- [PDOStatement::bindParam](#)
- [PDOStatement::bindValue](#)
- [PDOStatement::closeCursor](#)
- [PDOStatement::columnCount](#)
- [PDOStatement::errorCode](#)
- [PDOStatement::errorInfo](#)
- [PDOStatement::execute](#)
- [PDOStatement::fetch](#)
- [PDOStatement::fetchAll](#)
- [PDOStatement::fetchColumn](#)
- [PDOStatement::getAttribute](#)
- [PDOStatement::getColumnMeta](#)
- [PDOStatement::nextRowset](#)
- [PDOStatement::rowCount](#)
- [PDOStatement::setAttribute](#)
- [PDOStatement::setFetchMode](#)

8.101.13 PDO::commit() : Valide une transaction

bool PDO::commit (void)

[PDO::commit](#) valide une transaction, remet la connexion en mode autocommit en attendant l'appel à la fonction [PDO::beginTransaction](#) pour débiter une nouvelle transaction.

8.101.14 PDO::__construct() : Crée une instance PDO qui représente une connexion à une base de données

PDO PDO::__construct (string dsn , string username , string password , array driver_options)

[PDO::__construct](#) crée une instance PDO qui représente une connexion à la base de données désirée.

8.101.15 PDO::errorCode() : Retourne le SQLSTATE associé avec la dernière opération sur la base de données

string PDO::errorCode (void)

8.101.16 PDO::errorInfo() : Retourne les informations associées à l'erreur lors de la dernière opération sur la base de données

array PDO::errorInfo (void)

8.101.17 PDO::exec() : Exécute une requête SQL et retourne le nombre de lignes affectées

int PDO::exec (string statement)

PDO::exec exécute une requête SQL dans un appel d'une seule fonction, retourne le nombre de lignes affectées par la requête.

PDO::exec ne retourne pas de résultat pour une requête SELECT. Pour une requête SELECT dont vous auriez besoin une seule fois dans le programme, utilisez plutôt la fonction PDO::query . Pour une requête dont vous auriez besoin plusieurs fois, préparez un objet PDOStatement avec la fonction PDO::prepare et exécutez la requête avec la fonction PDOStatement::execute .

8.101.18 PDO::getAttribute() : Récupère un attribut d'une connexion à une base de données

mixed PDO::getAttribute (int attribute)

Cette fonction retourne la valeur d'un attribut d'une connexion à une base de données. Pour récupérer les attributs PDOStatement, référez-vous à la fonction PDOStatement::getAttribute .

Notez que quelques bases de données/drivers combinés ne supportent pas tous les attributs de connexion.

8.101.19 PDO::getAvailableDrivers() : Retourne un tableau des pilotes PDO disponibles

array PDO::getAvailableDrivers (void)

Cette fonction retourne tous les pilotes PDO présentement disponibles qui peuvent être utilisés dans le paramètre DSN de PDO::__construct . Il s'agit d'une méthode statique.

8.101.20 PDO::lastInsertId() : Retourne l'identifiant de la dernière ligne insérée ou la valeur d'une séquence

string PDO::lastInsertId (string name)

Retourne l'identifiant de la dernière ligne insérée, ou la dernière valeur d'une séquence d'objets, dépendamment du driver utilisé. Par exemple, PDO_PGSQL vous impose de spécifier le nom d'une séquence d'objet pour le paramètre name .

Note

Cette méthode peut ne pas retourner un résultat significatif suivant les drivers PDO utilisés, car la base de données utilisée peut ne pas supporter la notion de champs auto-incrémenté ou de séquences.

8.101.21 PDO::prepare() : Prépare une requête à l'exécution et retourne un objet

PDOStatement **PDO::prepare** (string *statement* , array *driver_options*)

Prépare une requête SQL à être exécutée par la méthode [PDOStatement::execute](#) . La requête SQL peut contenir zéro ou plusieurs noms (:name) ou marqueurs (?) pour lesquels les valeurs réelles seront substituées lorsque la requête sera exécutée. Vous ne pouvez pas utiliser les marqueurs nommés et les marqueurs interrogatifs dans une même requête SQL ; choisissez l'un ou l'autre.

Appeler [PDO::prepare](#) et [PDOStatement::execute](#) pour les requêtes qui doivent être exécutées plusieurs fois avec différentes valeurs de paramètres optimisent les performances de votre application en autorisant le driver à négocier coté client et/ou serveur avec le cache des requêtes et les méta-informations, et aident à prévenir les attaques par injection SQL en éliminant le besoin de protéger les paramètres manuellement.

PDO émulera les requêtes préparées / les paramètres liés pour les drivers qui ne le supportent pas nativement, et peut également réécrire les paramètres nommés ou les marqueurs en quelques choses de plus approprié, si le driver supporte un style mais pas l'autre.

8.101.22 PDO::query() : Exécute une requête SQL, retourne un jeu de résultats en tant qu'objet PDOStatement

PDOStatement **PDO::query** (string *statement*)

[PDO::query](#) exécute une requête SQL en appelant une seule fonction, retourne le jeu de résultats (s'il y en a) retourné par la requête en tant qu'objet PDOStatement.

Pour une requête que vous devez exécuter plusieurs fois, vous réaliserez de meilleurs performances si vous préparez l'objet PDOStatement en utilisant la fonction [PDO::prepare](#) et exécutez la requête via plusieurs appels à la fonction [PDOStatement::execute](#) .

Si vous ne récupérez pas toutes les données du jeux de résultats avant d'exécuter le prochain appel à [PDO::query](#) , votre appel peut échouer. Appeler [PDOStatement::closeCursor](#) pour libérer les ressources de la base de données associées à l'objet PDOStatement avant d'exécuter votre prochain appel à la fonction [PDO::query](#) .

8.101.23 PDO::quote() : Protège une chaîne pour l'utiliser dans une requête SQL PDO

string **PDO::quote** (string *string* , int *parameter_type*)

[PDO::quote](#) place des guillemets simples autour d'une chaîne d'entrées (si nécessaire) et échappe les caractères spéciaux présents dans la chaîne d'entrées, en utilisant le style de protection approprié au driver courant.

Si vous utilisez cette fonction pour construire des requêtes SQL, vous êtes **vivement** invité à utiliser `PDO::prepare` pour préparer les requêtes SQL avec des paramètres liées au lieu d'utiliser `PDO::quote` pour interpréter les entrées utilisateurs dans la requête SQL. Les requêtes préparées avec des paramètres liées ne sont pas seulement plus portables, plus maléables et plus sécurisées mais bien plus rapides à exécuter que d'interpréter les requêtes, vû que les côtés client et serveur peuvent mettre en cache une version compilée de la requête.

Tous les drivers PDO n'implémentent pas cette méthode (comme PDO_ODBC). Utilisez les requêtes préparées à la place.

8.101.24 PDO::rollback() : Annule une transaction

bool **PDO::rollback** (void)

Annule la transaction courante, initié par la fonction `PDO::beginTransaction` . C'est une erreur que d'appeler cette méthode s'il n'y a aucune transaction active.

Si la base de données est en mode autocommit, cette fonction restaurera le mode autocommit après l'annulation de la transaction.

Quelques bases de données, dont MySQL, exécuteront automatiquement un COMMIT lorsqu'une requête de définition de langage de base de données (DDL) comme DROP TABLE ou CREATE TABLE est exécutée dans une transaction. Ce COMMIT implicite vous empêchera d'annuler toutes autres modifications faites dans cette transaction.

8.101.25 PDO::setAttribute() : Définit un attribut

bool **PDO::setAttribute** (int attribute , mixed value)

`PDO::setAttribute` définit un attribut d'une connexion à une base de données. Les attributs génériques sont listés ci-dessous ; quelques drivers peuvent utiliser des attributs spécifiques.

- PDO::ATTR_CASE : Force les noms de colonnes en une casse spécifique.
 - ◆ PDO::CASE_LOWER : Force les noms de colonnes en minuscule.
 - ◆ PDO::CASE_NATURAL : Laisse les noms des colonnes en la casse définie par le driver de la base de données.
 - ◆ PDO::CASE_UPPER : Force les noms de colonnes en majuscule.
- PDO::ATTR_ERRMODE : Rapport des erreurs.
 - ◆ PDO::ERRMODE_SILENT : Définit juste les codes erreurs.
 - ◆ PDO::ERRMODE_WARNING : Lance un E_WARNING .
 - ◆ PDO::ERRMODE_EXCEPTION : Lance des exceptions .
- PDO::ATTR_ORACLE_NULLS (disponible pour tous les drivers, pas uniquement pour Oracle): Convertit les chaînes NULL et vides.
 - ◆ PDO::NULL_NATURAL : Aucune conversion.
 - ◆ PDO::NULL_EMPTY_STRING : Une chaîne vide est convertie en NULL .
 - ◆ PDO::NULL_TO_STRING : NULL est converti en une chaîne vide.
- PDO::ATTR_STRINGIFY_FETCHES : Convertit les valeurs numériques en chaînes de caractères lors de la récupération. Nécessite bool .
- PDO::ATTR_STATEMENT_CLASS : Définit la classe de la requête fournie par l'utilisateur dérivée de PDOStatement. Ne peut être utilisé avec des instances persistantes de PDO. Nécessite array(string classname, array(mixed constructor_args)) .

- `PDO::ATTR_AUTOCOMMIT` (disponible en OCI, Firebird et MySQL) : Valider automatiquement ou non chaque requête.
- `PDO::MYSQL_ATTR_USE_BUFFERED_QUERY` (disponible en MySQL) : Utilise les requêtes buffeurisées.

8.101.26 `PDOStatement::bindColumn()` : Lie une colonne à une variable PHP

`bool PDOStatement::bindColumn (mixed column , mixed param , int type)`

`PDOStatement::bindColumn` fait en sorte qu'une variable particulière soit liée à une colonne données dans le jeu de résultats depuis une requête. Chaque appel à la fonction `PDOStatement::fetch` ou `PDOStatement::fetchAll` mettera à jour toutes les variables qui sont liées aux colonnes.

Note

Vû que les informations sur les colonnes ne sont pas toujours disponibles à PDO tant que la requête n'est pas exécutée, les applications portables doivent appeler cette fonction **après** la fonction `PDO::execute` .

8.101.27 `PDOStatement::bindParam()` : Lie un paramètre à un nom de variable spécifique

`bool PDOStatement::bindParam (mixed parameter , mixed variable , int data_type , int length , mixed driver_options)`

Lie une variable PHP à un marquer nommé ou interrogatif correspondant dans une requête SQL utilisée pour préparer la requête. Contrairement à `PDOStatement::bindValue` , la variable est liée en tant que référence et ne sera évaluée qu'au moment de l'appel à la fonction `PDOStatement::execute` .

La plupart des paramètres sont des paramètres d'entrées, et sont utilisés en lecture seul pour construire la requête. Quelques drivers supportent l'invocation de procédures stockées qui retournent des données en tant que paramètres de sortie, et quelques autres en tant que paramètres entrées / sorties qui sont envoyés ensemble et sont mis à jour pour les recevoir.

8.101.28 `PDOStatement::bindValue()` : Associe une valeur à un paramètre

`bool PDOStatement::bindValue (mixed parameter , mixed value , int data_type)`

Associe une valeur à un nom correspondant ou à un point d'interrogation (comme paramètre fictif) dans la requête SQL qui fut utilisée pour préparer la requête.

8.101.29 `PDOStatement::closeCursor()` : Ferme le curseur, permettant à la requête d'être de nouveau exécutée

`bool PDOStatement::closeCursor (void)`

PDOStatement::closeCursor libère la connexion du serveur, permettant ainsi à d'autres requêtes SQL d'être exécutées, mais quitte la requête, permettant ainsi qu'elle soit de nouveau exécutée.

Cette méthode est utile pour les drivers de base de données qui ne supportent pas l'exécution d'objet PDOStatement lorsqu'un objet PDOStatement exécuté précédemment a encore des lignes non récupérées. Si votre driver de base de données souffre de cette limitation, le problème se manifestera de lui-même dans une erreur en dehors de la séquence.

PDOStatement::closeCursor est implémenté soit en tant que méthode optionnelle spécifique au driver (autorisant un maximum d'efficacité), ou en tant que fallback PDO générique si aucune fonction spécifique au driver n'est installée. Le fallback générique PDO est sémantiquement la même chose que d'écrire le code suivant dans votre script PHP :

```
<?php
do {
    while ($stmt->fetch())
        ;
    if (!$stmt->nextRowset())
        break;
} while (true);
```

8.101.30 PDOStatement::columnCount() : Retourne le nombre de colonnes dans le jeu de résultats

int **PDOStatement::columnCount** ()

Utilisez la fonction PDOStatement::columnCount pour retourner le nombre de colonnes dans le jeu de résultats représenté par l'objet PDOStatement.

Si l'objet PDOStatement a été retourné par la fonction PDO::query , le nombre de colonnes est immédiatement disponible.

Si l'objet PDOStatement a été retourné par la fonction PDO::prepare , un compte précis des colonnes ne sera pas disponible tant que vous n'invoquerez pas la fonction PDOStatement::execute

8.101.31 PDOStatement::errorCode() : Récupère le SQLSTATE associé lors de la dernière opération sur la requête

string **PDOStatement::errorCode** (void)

8.101.32 PDOStatement::errorInfo() : Récupère les informations sur l'erreur associée lors dernière opération sur la requête

array **PDOStatement::errorInfo** (void)

8.101.33 PDOStatement::execute() : Exécute une requête préparée

bool PDOStatement::execute (array input_parameters)

PDOStatement::execute exécute une requête préparée. Si la requête préparée inclue des marqueurs de positionnement, vous pouvez :

- appeler la fonction PDOStatement::bindParam pour lier les variables PHP aux marqueurs de positionnement : les variables liées passent leurs valeurs en entrée et reçoivent les valeurs de sortie s'il y en a, de leurs marqueurs de positionnement respectifs
- ou passer un tableau de valeurs de paramètres, uniquement en entrée

8.101.34 PDOStatement::fetch() : Récupère la ligne suivante d'un jeu de résultat PDO

mixed PDOStatement::fetch (int fetch_style , int cursor_orientation , int cursor_offset)

PDOStatement::fetch récupère une ligne depuis un jeu de résultats associé à l'objet PDOStatement . Le paramètre fetch_style détermine la façon dont PDO retourne la ligne.

8.101.35 PDOStatement::fetchAll() : Retourne un tableau contenant toutes les lignes du jeu d'enregistrements

array PDOStatement::fetchAll (int fetch_style , int column_index)

8.101.36 PDOStatement::fetchColumn() : Retourne une colonne depuis la ligne suivante d'un jeu de résultats

string PDOStatement::fetchColumn (int column_number)

Retourne une colonne depuis la ligne suivante d'un jeu de résultats.

8.101.37 PDOStatement::getAttribute() : Récupère un attribut de requête

mixed PDOStatement::getAttribute (int attribute)

Récupère un attribut de la requête. Actuellement, aucun attribut générique n'existe, mais uniquement des spécificités du driver :

- PDO::ATTR_CURSOR_NAME (spécificité de Firebird et d'ODBC) : Récupère le nom du curseur pour UPDATE ... WHERE CURRENT OF .

8.101.38 PDOStatement::getColumnMeta() : Retourne les méta-données pour une colonne d'un jeu de résultats

mixed PDOStatement::getColumnMeta (int column)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Récupère les méta-informations pour une colonne d'un jeu de résultat dans un tableau associatif.

Attention

Tous les drivers PDO ne supportent pas la fonction <code>PDOStatement::getColumnMeta</code> .

8.101.39 PDOStatement::nextRowset() : Avance à la prochaine ligne de résultats d'un gestionnaire de lignes de résultats multiples

bool PDOStatement::nextRowset (void)

Quelques bases de données supportent les procédures stockées qui retournent plus d'une ligne de résultats (aussi connu comme des jeux de résultats). `PDOStatement::nextRowSet` vous permet d'accéder à la seconde et suivantes lignes de résultats associées avec l'objet PDOStatement. Chaque lignes de résultats ont des jeux différents de colonnes depuis la ligne de résultats.

8.101.40 PDOStatement::rowCount() : Retourne le nombre de lignes affectées par le dernier appel à la fonction PDOStatement::execute()

int PDOStatement::rowCount (void)

`PDOStatement::rowCount` retourne le nombre de lignes affectées par la dernière requête DELETE, INSERT ou UPDATE exécutée par l'objet PDOStatement correspondant.

Si la dernière requête SQL exécutée par l'objet PDOStatement associé est une requête de type SELECT, quelques bases de données retourneront le nombre de lignes retournées par cette requête. Néanmoins, ce comportement n'est pas garanti pour toutes les bases de données et ne devrait pas être exécuté pour des applications portables.

8.101.41 PDOStatement::setAttribute() : Définit un attribut de requête

bool PDOStatement::setAttribute (int attribute , mixed value)

Définit un attribut de la requête. Actuellement, aucun attribut générique n'est défini, mais uniquement des spécificités du driver :

- PDO::ATTR_CURSOR_NAME (spécificité de Firebird et d'ODBC) : Définit le nom du curseur pour UPDATE ... WHERE CURRENT OF .

8.101.42 PDOStatement::setFetchMode() : Définit le mode de récupération par défaut pour cette requête

bool PDOStatement::setFetchMode (int mode)

8.102 Paiement par Verisign

8.102.1 Introduction

Cette extension vous permet d'effectuer des transactions avec des cartes de crédit en utilisant les services Verisign Payment Services, anciennement connu sous le nom de Signio (<http://www.verisign.com/products-services/payment-processing/online-payment/payflow-pro/index.html>).

Lorsque vous utilisez ces fonctions, vous pouvez omettre d'appeler les fonctions `pfpro_init` et `pfpro_cleanup` : l'extension se chargera de le faire automatiquement. Cependant, elles sont toujours disponibles au cas où vous auriez un grand nombre de transactions à traiter, ou que vous souhaiteriez un contrôle plus fin de la bibliothèque. Vous pouvez effectuer autant de transactions que vous le souhaitez avec `pfpro_process` lors d'une connexion.

Ces fonctions ont été ajoutées en PHP 4.0.2.

Note
Ces fonctions ne font que fournir un accès aux services Verisign Payment Services. Assurez-vous bien de lire le "Payflow Pro Developers Guide" pour plus de détails sur les paramètres.
Note
Cette extension a été déplacée dans le module <code>PECL</code> et ne sera plus intégrée dans PHP à partir de PHP 5.1.0.
Note
Cette extension n'est pas disponible sur les plates-formes Windows.

8.102.2 Pré-requis

Vous devez aussi utiliser le SDK approprié sur votre plate-forme : il est disponible [l'interface du manager](#), une fois que vous vous êtes inscrit.

Une fois que vous avez téléchargé le SDK vous devez copier les fichiers depuis le dossier lib de la distribution. Copiez le fichier d'en-têtes `pfpro.h` dans `/usr/local/include` et la bibliothèque `libpfpro.so` dans `/usr/local/lib`.

Alternativement, vous pouvez extraire l'archive depuis Verisign dans un dossier et faites-lui référence durant la compilation avec l'option `--with-pfpro[=DIR]` :

Configuration explicite
<pre>tar -zxf pfpro_sunsparc.tar.gz -C /usr/local/ ./configure --with-pfpro=/usr/local/verisign/payflowpro/sunsparc</pre>
Note
La dernière portion du chemin spécifiée dans l'exemple ci-dessus, dans ce cas <code>sunsparc</code> , est basé sur l'architecture pour lequel votre Verisign SDK a été compilé.

8.102.3 Installation

Ces fonctions ne sont disponibles que si PHP a été compilé avec l'option `--with-pfpro[=DIR]`.

Attention
Si vous désirez utiliser cette extension avec l'extension OpenSSL ou avec ModSSL, vous devez la compiler en tant que partagée : <code>--with-pfpro=shared,/usr/local</code> .

8.102.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable	Historique
<code>pfpro.defaulthost/PFPRO_VERSION < 3</code>	"test.signio.com"	PHP_INI_ALL	
<code>pfpro.defaulthost</code>	"test-payflow.verisign.com"	PHP_INI_ALL	Disponible depuis PHP 4.0.2.
<code>pfpro.defaultport</code>	"443"	PHP_INI_ALL	Disponible depuis PHP 4.0.2.
<code>pfpro.defaulttimeout</code>	"30"	PHP_INI_ALL	Disponible depuis PHP 4.0.2.
<code>pfpro.proxyaddress</code>	""	PHP_INI_ALL	Disponible depuis PHP 4.0.2.
<code>pfpro.proxyport</code>	""	PHP_INI_ALL	Disponible depuis PHP 4.0.2.
<code>pfpro.proxylogon</code>	""	PHP_INI_ALL	Disponible depuis PHP 4.0.2.
<code>pfpro.proxypassword</code>	""	PHP_INI_ALL	Disponible depuis PHP 4.0.2.

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du `php.ini`](#) .

8.102.5 Types de ressources

Cette extension ne définit aucune ressource.

8.102.6 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [pfpro_cleanup](#)
- [pfpro_init](#)
- [pfpro_process_raw](#)
- [pfpro_process](#)
- [pfpro_version](#)

`pfpro_process` effectue une transaction avec Payflow Pro. Le premier paramètre `parameters` est un tableau associatif contenant des paires clés/valeurs, qui seront encodées, puis passées au serveur.

Le second paramètre `address` indique quel hôte contacter. Il est optionnel. Par défaut, il vaut "test.signio.com" : vous devrez probablement le remplacer par "connect.signio.com" pour effectuer de vraies transactions.

Le troisième paramètre `port` spécifie le port de connexion. Par défaut, c'est 443, le port SSL standard.

Le quatrième paramètre `timeout` indique le temps de timeout à utiliser. Par défaut, c'est 30 secondes. Notez que ce timeout ne prend effet que lorsqu'une connexion a été établie avec un serveur : votre script peut potentiellement attendre indéfiniment en cas de problèmes DNS ou réseau.

Le cinquième paramètre `proxy_address` indique le nom du proxy SSL. Le sixième paramètre `proxy_port` indique le port à utiliser sur ce proxy.

Les septième et huitième paramètres, `proxy_logon` et `proxy_password` indiquent le nom de compte et le mot de passe à utiliser sur le proxy.

`pfpro_process` retourne un tableau associatif avec les clés/valeurs de la réponse.

Note
Lisez attentivement le "Payflow Pro Developers Guide" pour connaître les détails des autres paramètres.
Exemple de transaction Payflow Pro
<pre><?php pfpro_init(); \$transaction = array('USER' => 'monlogin', 'PWD' => 'monmotdepasse', 'PARTNER' => 'VeriSign', 'TRXTYPE' => 'S', 'TENDER' => 'C', 'AMT' => 1.50, 'ACCT' => '4111111111111111', 'EXPDATE' => '0904'); \$response = pfpro_process(\$transaction); if (!\$response) { die("Impossible d'établir un lien avec Verisign.\n"); } echo "La réponse de Verisign était ", \$response['RESULT'], "\n"; echo "C'est-à-dire : ", \$response['RESPMSG'], "\n"; echo "\nLa requête de transaction: "; print_r(\$transaction); echo "\nLa réponse: "; print_r(\$response); pfpro_cleanup(); ?></pre>

8.102.11 pfpro_version() : Retourne le numéro de version de Payflow Pro

string **pfpro_version** (void)

pfpro_version retourne la version de la bibliothèque Payflow Pro. Au moment de rédaction de ce document, le numéro était L211.

8.103 PostgreSQL

8.103.1 Introduction

La base de données PostgreSQL est un produit Open Source, disponible sans frais. PostgreSQL, développé au département de Science informatique, à UC Berkeley, met en place la majorité des concepts des bases relationnelles actuellement disponibles sur le marché. PostgreSQL accepte le langage SQL92/SQL3, assure l'intégrité transactionnelle et l'extension de type. PostgreSQL est une évolution du code original de Berkeley.

8.103.2 Pré-requis

Pour accéder au support PostgreSQL, vous avez besoin de PostgreSQL 6.5 ou plus récent ; PostgreSQL 7.0 ou plus récent pour activer toutes les fonctionnalités du module PostgreSQL. PostgreSQL supporte de nombreux jeux de caractères, y compris les jeux multi-octets asiatiques. La version courante et plus de détails sur PostgreSQL sont accessibles sur le site <http://www.postgresql.org/> et la [Documentation PostgreSQL](#) .

8.103.3 Installation

Afin d'activer le support PostgreSQL, l'option `--with-pgsql[=DIR]` est nécessaire lors de la compilation de PHP. DIR est le chemin du dossier d'installation de PostgreSQL, et par défaut il vaut `/usr/local/pgsql` . Si le module de chargement dynamique est disponible, le module PostgreSQL peut être chargé avec la directive `extension` du fichier `php.ini` ou via la fonction `dl` .

8.103.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable	Historique
<code>pgsql.allow_persistent</code>	"1"	PHP_INI_SYSTEM	
<code>pgsql.max_persistent</code>	"-1"	PHP_INI_SYSTEM	
<code>pgsql.max_links</code>	"-1"	PHP_INI_SYSTEM	
<code>pgsql.auto_reset_persistent</code>	"0"	PHP_INI_SYSTEM	Disponible depuis PHP 4.2.0.
<code>pgsql.ignore_notice</code>	"0"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
<code>pgsql.log_notice</code>	"0"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

`pgsql.allow_persistent` booléen

Autorise ou non les connexions persistantes Postgre.

`pgsql.max_persistent` entier

Le nombre maximum de connexions persistantes Postgre par processus.

`pgsql.max_links` entier

Le nombre maximum de connexions Postgre par processus, y compris les connexions persistantes.

`pgsql.auto_reset_persistent` [entier](#)

Détecte les connexions persistantes (ouvertes avec `pg_pconnect`) rompues.

`pgsql.ignore_notice` [entier](#)

Active ou non l'affichage des notices PostgreSQL.

`pgsql.log_notice` [entier](#)

Active ou non l'enregistrement en fichier log des messages notices PostgreSQL. La directive PHP `pgsql.ignore_notice` doit être désactivée pour pouvoir enregistrer les messages de notices en fichier log.

8.103.5 Types de ressources

Il y a deux types de ressource utilisé dans le module PostgreSQL. La première est l'identifiant pour la connexion à la base de données et le second est une ressource qui contient le résultat d'une requête.

8.103.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

`PGSQL_ASSOC` ([entier](#))

Passée à `pg_fetch_array` . Retourne un tableau associatif des noms et valeurs des champs.

`PGSQL_NUM` ([entier](#))

Passée à `pg_fetch_array` . Retourne un tableau à index numérique des numéros et valeurs des champs.

`PGSQL_BOTH` ([entier](#))

Passée à `pg_fetch_array` . Retourne un tableau des valeurs des champs qui est indexés numériquement (par le numéro des champs) et indexés associativement (par le nom des champs).

`PGSQL_CONNECT_FORCE_NEW` ([entier](#))

Passée à `pg_connect` pour forcer la création d'une nouvelle connexion, plutôt que de réutiliser une connexion identique existante.

`PGSQL_CONNECTION_BAD` ([entier](#))

Retournée par `pg_connection_status` indiquant que la connexion à la base de données est invalide.

`PGSQL_CONNECTION_OK` ([entier](#))

Retournée par `pg_connection_status` indiquant que la connexion à la base de données est valide.

`PGSQL_SEEK_SET` ([entier](#))

Passée à `pg_lo_seek` . Le positionnement commencera au début de l'objet.

`PGSQL_SEEK_CUR` ([entier](#))

Passée à `pg_lo_seek` . Le positionnement commencera à la position courante.

`PGSQL_SEEK_END` ([entier](#))

Passée à `pg_lo_seek` . Le positionnement commencera à la fin de l'objet.

`PGSQL_EMPTY_QUERY` ([entier](#))

Retournée par `pg_result_status` . La chaîne de caractères envoyée au serveur était vide.

`PGSQL_COMMAND_OK` ([entier](#))

Retournée par `pg_result_status` . Commande correctement complétée ne retournant aucune donnée.

`PGSQL_TUPLES_OK` ([entier](#))

- Retournée par pg_result_status . Commande correctement complétée retournant des données (comme SELECT ou SHOW) .
- PGSQL_COPY_OUT** (entier)
Retournée par pg_result_status . Copie (à partir du serveur) de transfert de données commencée.
- PGSQL_COPY_IN** (entier)
Retournée par pg_result_status . Copie (vers le serveur) de transfert de données commencée.
- PGSQL_BAD_RESPONSE** (entier)
Retournée par pg_result_status . La réponse du serveur n'a pas été comprise.
- PGSQL_NONFATAL_ERROR** (entier)
Retournée par pg_result_status . Une erreur non fatale (de niveau notice ou warning) s'est produite.
- PGSQL_FATAL_ERROR** (entier)
Retournée par pg_result_status . Une erreur fatale s'est produite.
- PGSQL_TRANSACTION_IDLE** (entier)
Retournée par pg_transaction_status . La connexion est présentement libre, aucune transaction en cours.
- PGSQL_TRANSACTION_ACTIVE** (entier)
Retournée par pg_transaction_status . Une commande est en cours sur la connexion. Une requête a été envoyée sur la connexion et n'a toujours pas été complétée.
- PGSQL_TRANSACTION_INTRANS** (entier)
Retournée par pg_transaction_status . La connexion est libre, dans un bloc de transaction.
- PGSQL_TRANSACTION_INERROR** (entier)
Retournée par pg_transaction_status . La connexion est libre, dans un bloc de transaction échoué.
- PGSQL_TRANSACTION_UNKNOWN** (entier)
Retournée par pg_transaction_status . La connexion est mauvaise.
- PGSQL_DIAG_SEVERITY** (entier)
Passée à pg_result_error_field . La sévérité; le contenu du champ est ERROR , FATAL ou PANIC (dans un message d'erreur) ou WARNING , NOTICE , DEBUG , INFO ou LOG (dans un message d'avertissement) ou une traduction localisée parmi celles-ci. Toujours présent.
- PGSQL_DIAG_SQLSTATE** (entier)
Passée à pg_result_error_field . Le code SQLSTATE pour cette erreur. Le code SQLSTATE identifie le type d'erreur qui s'est produite; cela peut être utilisé par des applications d'entrée pour effectuer des opérations spécifiques (comme la gestion d'erreur) en réponse à une erreur de base de données particulière. Ce champ ne peut être localisé et est toujours présent.
- PGSQL_DIAG_MESSAGE_PRIMARY** (entier)
Passée à pg_result_error_field . Le champ d'erreur primaire interprétable pour l'utilisateur (normalement une ligne). Toujours présent.
- PGSQL_DIAG_MESSAGE_DETAIL** (entier)
Passée à pg_result_error_field . Détail : un second optionnel message d'erreur apportant plus de détails à propos du problème. Peut être sur plusieurs lignes.
- PGSQL_DIAG_MESSAGE_HINT** (entier)
Passée à pg_result_error_field . Conseil : une suggestion optionnelle qui indique que faire à propos du problème. Ceci est prévu d'être différent de l'erreur puisqu'elle offre un conseil (potentiellement inadéquat) plutôt que les faits véridiques. Peut être sur plusieurs lignes.
- PGSQL_DIAG_STATEMENT_POSITION** (entier)
Passée à pg_result_error_field . Une chaîne de caractères contenant une valeur

entière décimale indiquant une erreur de position du curseur en tant qu'index dans la requête originale. Le premier caractère a l'index 1 et les positions sont mesurées en caractères, non en octets.

PGSQL_DIAG_INTERNAL_POSITION (entier)

Passée à pg_result_error_field . Ceci est défini étant la même chose que le champ PG_DIAG_STATEMENT_POSITION , mais cela est utilisé lorsque la position du curseur réfère à une commande générée internement plutôt que d'une envoyée par le client. Le champ PG_DIAG_INTERNAL_QUERY apparaîtra toujours lorsque ce champ apparaît.

PGSQL_DIAG_INTERNAL_QUERY (entier)

Passée à pg_result_error_field . Le texte d'une commande générée internement échouée. Cela peut être, par exemple, une requête SQL envoyée par une fonction PL/pgSQL.

PGSQL_DIAG_CONTEXT (entier)

Passée à pg_result_error_field . Une indication du contexte dans lequel l'erreur s'est produit. Présentement, ceci inclue une pile d'appel des traceback des fonctions procédurales actives ainsi que des requête générées à l'interne. Le traçage est une entrée par ligne, les plus récentes en premier.

PGSQL_DIAG_SOURCE_FILE (entier)

Passée à pg_result_error_field . Le nom du fichier de l'emplacement du code source PostgreSQL où l'erreur a été reportée.

PGSQL_DIAG_SOURCE_LINE (entier)

Passée à pg_result_error_field . Le nombre de ligne de l'emplacement du code source PostgreSQL où l'erreur a été reportée.

PGSQL_DIAG_SOURCE_FUNCTION (entier)

Passée à pg_result_error_field . Le nom de la fonction de source code PostgreSQL reportant l'erreur.

PGSQL_ERRORS_TERSE (entier)

Passée à pg_set_error_verbosity . Spécifie que les messages retournés incluent la sévérité, le texte primaire ainsi que la position seulement; ceci devrait entrer sur une seule ligne.

PGSQL_ERRORS_DEFAULT (entier)

Passée à pg_set_error_verbosity . Le mode par défaut produit des messages qui incluent ce qui est plus haut et des détails en plus, conseil ou des champs contextes (ceci peut être sur plusieurs lignes).

PGSQL_ERRORS_VERBOSE (entier)

Passée à pg_set_error_verbosity . Le mode verbeux inclue tous les champs disponibles.

PGSQL_STATUS_LONG (entier)

Passée à pg_result_status . Indique que le code résultat est désiré numérique.

PGSQL_STATUS_STRING (entier)

Passée à pg_result_status . Indique que le tag de résultat de commande est désiré textuel.

PGSQL_CONV_IGNORE_DEFAULT (entier)

Passée à pg_convert . Ignore les valeurs par défaut dans la table pendant la conversion.

PGSQL_CONV_FORCE_NULL (entier)

Passée à pg_convert . Utilise NULL à la place d'une chaîne de caractères vide.

PGSQL_CONV_IGNORE_DEFAULT (entier)

Passée à pg_convert . Ignore la conversion de NULL à l'intérieur des colonnes NOT NULL .

8.103.7 Notes

Note

Les fonctions ne sont pas toutes supportées par toutes les versions. Cela dépend de votre version de libpq (La bibliothèque cliente de PostgreSQL C) et comment libpq est compilé. Si les extensions PHP PostgreSQL sont manquantes, alors c'est parce que votre version de libpq ne les supporte pas.

Note

La plupart des fonctions PostgreSQL acceptent le paramètre `connection` comme premier paramètre optionnel. S'il n'est pas fourni, la dernière connexion ouverte est utilisée. Si elle n'existe pas, les fonctions retournent `FALSE`.

Note

PostgreSQL transforme automatiquement tous les identifiants (par exemple, les noms de tables/colonnes) en minuscule à la création d'objet et lors de la requête. Pour forcer l'utilisation des caractères majuscules ou minuscules et majuscules des identifiants, vous devez échapper l'identifiant en utilisant des guillemets ("").

Note

PostgreSQL n'a pas de commande spéciale pour récupérer les schémas des bases de données (par exemple, toutes les tables dans la base de données). À la place, il y a un schéma standard nommé `information_schema` dans PostgreSQL 7.4 et supérieure contenant les vues du système avec toutes les informations nécessaires, dans un format de requête facile. Voyez la [Documentation PostgreSQL](#) pour tous les détails.

8.103.8 Exemples

Ce simple exemple montre comment se connecter, exécuter une requête et afficher les lignes résultantes et se déconnecter d'une base de données PostgreSQL.

Exemple général de l'extension PostgreSQL

```
<?php
// Connexion, sélection de la base de données
$dbconn = pg_connect("host=localhost dbname=publishing user=www password=foo")
    or die('Connexion impossible : ' . pg_last_error());

// Exécution de la requête SQL
$query = 'SELECT * FROM auteurs';
$result = pg_query($query) or die('Échec requête : ' . pg_last_error());

// Affichage des résultats en HTML
echo "<table>\n";
while ($line = pg_fetch_array($result, null, PGSQL_ASSOC)) {
    echo "\t<tr>\n";
    foreach ($line as $col_value) {
        echo "\t\t<td>$col_value</td>\n";
    }
    echo "\t</tr>\n";
}
echo "</table>\n";

// Libère le resultatset
pg_free_result($result);

// Ferme la connexion
pg_close($dbconn);
?>
```

- [Introduction](#)

- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Notes](#)
- [Exemples](#)
- [pg_affected_rows](#)
- [pg_cancel_query](#)
- [pg_client_encoding](#)
- [pg_close](#)
- [pg_connect](#)
- [pg_connection_busy](#)
- [pg_connection_reset](#)
- [pg_connection_status](#)
- [pg_convert](#)
- [pg_copy_from](#)
- [pg_copy_to](#)
- [pg_dbname](#)
- [pg_delete](#)
- [pg_end_copy](#)
- [pg_escape_bytea](#)
- [pg_escape_string](#)
- [pg_execute](#)
- [pg_fetch_all_columns](#)
- [pg_fetch_all](#)
- [pg_fetch_array](#)
- [pg_fetch_assoc](#)
- [pg_fetch_object](#)
- [pg_fetch_result](#)
- [pg_fetch_row](#)
- [pg_field_is_null](#)
- [pg_field_name](#)
- [pg_field_num](#)
- [pg_field_prtlen](#)
- [pg_field_size](#)
- [pg_field_type_oid](#)
- [pg_field_type](#)
- [pg_free_result](#)
- [pg_get_notify](#)
- [pg_get_pid](#)
- [pg_get_result](#)
- [pg_host](#)
- [pg_insert](#)
- [pg_last_error](#)
- [pg_last_notice](#)
- [pg_last_oid](#)
- [pg_lo_close](#)
- [pg_lo_create](#)
- [pg_lo_export](#)
- [pg_lo_import](#)
- [pg_lo_open](#)
- [pg_lo_read_all](#)
- [pg_lo_read](#)

- [pg_lo_seek](#)
- [pg_lo_tell](#)
- [pg_lo_unlink](#)
- [pg_lo_write](#)
- [pg_meta_data](#)
- [pg_num_fields](#)
- [pg_num_rows](#)
- [pg_options](#)
- [pg_parameter_status](#)
- [pg_pconnect](#)
- [pg_ping](#)
- [pg_port](#)
- [pg_prepare](#)
- [pg_put_line](#)
- [pg_query_params](#)
- [pg_query](#)
- [pg_result_error_field](#)
- [pg_result_error](#)
- [pg_result_seek](#)
- [pg_result_status](#)
- [pg_select](#)
- [pg_send_execute](#)
- [pg_send_prepare](#)
- [pg_send_query_params](#)
- [pg_send_query](#)
- [pg_set_client_encoding](#)
- [pg_set_error_verbosity](#)
- [pg_trace](#)
- [pg_transaction_status](#)
- [pg_tty](#)
- [pg_unescape_bytea](#)
- [pg_untrace](#)
- [pg_update](#)
- [pg_version](#)

8.103.10 [pg_cancel_query\(\)](#) : Annule une requête asynchrone

bool [pg_cancel_query](#) (resource connection)

[pg_cancel_query](#) annule la requête asynchrone, démarrée avec [pg_send_query](#) , [pg_send_query_params](#) ou [pg_send_execute](#) . Vous ne pouvez pas annuler une requête démarrée avec [pg_query](#) .

8.103.11 [pg_client_encoding\(\)](#) : Lit l'encodage du client

string [pg_client_encoding](#) (*resource connection*)

PostgreSQL supporte la conversion automatique entre le serveur et le client pour certains jeux de caractères. [pg_client_encoding](#) retourne l'encodage du client. La chaîne de retour sera un des encodages standards de PostgreSQL.

Note

Cette fonction requiert PHP 4.0.3 ou plus récent et PostgreSQL version 7.0 ou plus récent. Si la bibliothèque libpq est compilée sans le support de l'encodage multi-octets, `pg_client_encoding` retournera toujours `SQL_ASCII`. Le support de l'encodage dépend de la version de PostgreSQL. Référez-vous à la documentation de PostgreSQL sur les encodages supportés.

Auparavant, cette fonction s'appelait `pg_clientencoding`.

8.103.12 `pg_close()` : Termine une connexion PostgreSQL

bool `pg_close` (*resource connection*)

`pg_close` ferme la connexion au serveur PostgreSQL associé à `connection`.

Note

Il n'est généralement pas nécessaire de fermer une connexion non persistante, car elles sont automatiquement fermées à la fin d'un script.

Si des objets de grande taille ont été ouverts avec cette connexion, ne fermez pas la connexion avant d'avoir refermé les objets.

8.103.13 `pg_connect()` : Établit une connexion PostgreSQL

resource `pg_connect` (string *connection_string* , int *connect_type*)

`pg_connect` ouvre une connexion à une base de données PostgreSQL grâce à la chaîne de connexion `connection_string`.

Si un deuxième appel à `pg_connect` est fait avec les mêmes arguments, aucune nouvelle connexion ne sera établie tant que vous passerez `PGSQL_CONNECT_FORCE_NEW` à `connect_type`, mais la connexion précédente sera retournée.

L'ancienne syntaxe `$conn = pg_connect("host", "port", "options", "tty", "dbname")` est obsolète.

8.103.14 `pg_connection_busy()` : Vérifie si la connexion PostgreSQL est occupée

bool `pg_connection_busy` (resource *connection*)

`pg_connection_busy` détermine si la connexion est occupée. Si elle est occupée, une requête a déjà été lancée, et est en cours. Si `pg_get_result` est utilisée, elle sera alors bloquée.

8.103.15 `pg_connection_reset()` : Relance la connexion au serveur PostgreSQL

bool `pg_connection_reset` (resource *connection*)

`pg_connection_reset` effectue une reconnexion au serveur, avec les mêmes paramètres que lors de la connexion précédente avec `connection`. Cette fonction est pratique pour le traitement des erreurs.

8.103.16 `pg_connection_status()` : Lit le statut de la connexion PostgreSQL

int `pg_connection_status` (resource connection)

`pg_connection_status` retourne le statut de la connexion `connection` .

8.103.17 `pg_convert()` : Convertit des tableaux associatifs en une commande PostgreSQL

array `pg_convert` (resource connection , string table_name , array assoc_array , int options)

`pg_convert` vérifie et convertit le tableau associatif `assoc_array` en une requête SQL valide. Pour que `pg_convert` fonctionne, il faut que la table `table_name` existe, et contienne au moins autant de colonnes que le tableau `assoc_array` a d'éléments. Les noms des champs de `table_name` doivent correspondre aux index du tableau dans `assoc_array` . Retourne un tableau avec les valeurs converties en cas de succès, et sinon, FALSE .

Note
S'il y a des champs booléens dans <code>table_name</code> , n'utilisez pas la constante TRUE dans <code>assoc_array</code> . Elle sera convertie en chaîne de caractères 'TRUE' qui ne sera pas une entrée valide pour un champ booléen en PostgreSQL. Utilisez une de ces valeurs à la place : t, true, 1, y, yes.
Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

8.103.18 `pg_copy_from()` : Insère des lignes dans une table à partir d'un tableau

bool `pg_copy_from` (resource connection , string table_name , array rows , string delimiter , string null_as)

`pg_copy_from` insère les éléments du tableau `rows` dans une table. Cette fonction utilise la commande SQL interne COPY FROM .

8.103.19 `pg_copy_to()` : Copie une table dans un tableau

array `pg_copy_to` (resource connection , string table_name , string delimiter , string null_as)

`pg_copy_to` copie la table `table_name` dans un tableau. Cette fonction utilise la commande interne SQL COPY TO pour insérer les tableaux.

8.103.20 `pg_dbname()` : Retourne le nom de la base de données PostgreSQL

string `pg_dbname` (resource connection)

`pg_dbname` retourne le nom de la base de données PostgreSQL associée à l'index de connexion `connection` .

8.103.21 `pg_delete()` : Efface des lignes PostgreSQL

mixed `pg_delete` (resource `connection` , string `table_name` , array `assoc_array` , int `options`)

`pg_delete` efface les lignes de la table `table_name` , spécifiées par le tableau associatif `assoc_array` . Si option est spécifié, `pg_convert` est appliqué à `assoc_array` avec cette option.

8.103.22 `pg_end_copy()` : Synchronise avec le serveur PostgreSQL

bool `pg_end_copy` (resource `connection`)

`pg_end_copy` synchronise le client PostgreSQL (normalement un processus serveur web) avec le serveur PostgreSQL, après une opération de copie faite par `pg_put_line` . `pg_end_copy` doit être utilisé, autrement le serveur PostgreSQL ne sera plus synchronisé avec le client et émettra une erreur.

8.103.23 `pg_escape_bytea()` : Protège une chaîne pour insertion dans un champ bytea

string `pg_escape_bytea` (string `data`)

`pg_escape_bytea` protège les caractères de la chaîne `data` avec le mode `bytea`. La chaîne protégée est retournée.

Note

Lorsque vous utilisez une commande `SELECT` avec des données de type `bytea`, PostgreSQL retourne des valeurs octales, préfixées avec des anti-slash `\` (par exemple `\032`). Les utilisateurs doivent effectuer la conversion en format binaire manuellement.

`pg_escape_bytea` requiert PostgreSQL 7.2 ou plus récent. Avec PostgreSQL 7.2.0 et 7.2.1, les données `bytea` doivent être transtypée lorsque vous activez le support des chaînes de caractères multi-octets. C'est-à-dire `INSERT INTO test_table (image) VALUES ('$image_escaped'::bytea);` . PostgreSQL 7.2.2 ou plus récent ne requiert pas cette manipulation. Toutefois, si le client et le serveur n'utilisent pas le même jeu de caractères, il peut arriver des erreurs. Il faut alors forcer le transtypage manuellement.

8.103.24 `pg_escape_string()` : Protège une chaîne de caractères pour l'insérer dans un champ texte

string `pg_escape_string` (string `data`)

`pg_escape_string` protège une chaîne de caractères pour l'insérer dans la base de données. Elle retourne la chaîne ainsi protégée au format PostgreSQL. L'utilisation de cette fonction est recommandée, à la place de `addslashes` . Si le type de la colonne est `bytea`, `pg_escape_bytea` doit plutôt être utilisée.

Note

Cette fonction requiert PostgreSQL 7.2 ou plus récent.
--

8.103.25 `pg_execute()` : Envoie une requête pour exécuter une requête préparée avec les paramètres donnés et attend le résultat

resource `pg_execute` (resource connection , string stmtname , array params) resource `pg_execute` (string stmtname , array params)

Envoie une requête pour exécuter une requête préparée avec les paramètres donnés et attend le résultat.

`pg_execute` est comme `pg_query_params` , mais la commande qui sera exécutée est spécifiée en nommant une requête préalablement préparée, au lieu de donner une chaîne en tant que requête. Cette caractéristique permet aux commandes qui seront utilisées à plusieurs reprises d'être analysées et planifiées qu'une seule fois, plutôt que d'être exécutées chaque fois. La requête doit avoir été préalablement préparée dans la session courante. `pg_execute` est supportée seulement avec les versions PostgreSQL 7.4 ou plus récentes; la commande échouera si vous l'utilisez avec des versions antérieures.

Les paramètres sont identiques à la fonction `pg_query_params` à l'exception du nom de la requête préparée qui est donnée à la place de la requête sous forme de chaîne.

8.103.26 `pg_fetch_all_columns()` : Récupère toutes les lignes d'une colonne de résultats particulière en tant que tableau

array `pg_fetch_all_columns` (resource result , int column)

`pg_fetch_all_columns` retourne un tableau qui contient toutes les lignes (enregistrements) d'une colonne particulière d'une ressource de résultats.

Note

Cette fonction définit les champs NULL à la valeur PHP NULL .

8.103.27 `pg_fetch_all()` : Lit toutes les lignes d'un résultat

array `pg_fetch_all` (resource result)

`pg_fetch_all` retourne un tableau qui contient toutes les lignes du résultat result .

Note

Cette fonction définit les champs NULL à la valeur PHP NULL .

8.103.28 `pg_fetch_array()` : Lit une ligne de résultat PostgreSQL dans un tableau

array `pg_fetch_array` (resource result , int row , int result_type)

`pg_fetch_array` retourne un tableau qui contient la ligne demandée.

`pg_fetch_array` est une version évoluée de `pg_fetch_row`. En plus de proposer un tableau à indice numérique, elle peut aussi enregistrer les données dans un tableau associatif, en utilisant les noms des champs comme clés. Ces deux fonctions utilisent le tableau associatif par défaut.

Note
Cette fonction définit les champs NULL à la valeur PHP NULL .
<code>pg_fetch_array</code> n'est pas significativement plus lente que <code>pg_fetch_row</code> et elle apporte un confort d'utilisation appréciable.

Note
<code>row</code> est devenu optionnel depuis PHP 4.1.0.
Note
<code>result_type</code> a été ajouté en PHP 4.0.0.

8.103.29 `pg_fetch_assoc()` : Lit une ligne de résultat PostgreSQL sous forme de tableau numérique

array `pg_fetch_assoc` (resource result , int row)

`pg_fetch_assoc` retourne un tableau associatif qui contient la ligne en cours dans le résultat result .

`pg_fetch_assoc` est équivalent d'appeler `pg_fetch_row` avec `PGSQL_ASSOC` comme troisième paramètre (qui est optionnel). Cela retournera seulement un tableau associatif. Si vous avez besoin d'indices numériques, utilisez `pg_fetch_row` .

Note
Cette fonction définit les champs NULL à la valeur PHP NULL .
<code>pg_fetch_assoc</code> n'est pas significativement plus lente que <code>pg_fetch_row</code> et elle apporte un confort d'utilisation appréciable.

Note
<code>row</code> est devenu optionnel depuis PHP 4.1.0.

8.103.30 `pg_fetch_object()` : Lit une ligne de résultat PostgreSQL dans un objet

object `pg_fetch_object` (resource result , int row , int result_type) object `pg_fetch_object` (resource result , int row , string class_name , array params)

`pg_fetch_object` retourne un objet ainsi que ses propriétés qui correspond aux noms des champs de la ligne. La fonction peut optionnellement instancier un objet d'une classe spécifique et passer les paramètres au constructeur de cette classe.

Note
Cette fonction définit les champs NULL à la valeur PHP NULL .
Du point de vue vitesse, la fonction est identique à <code>pg_fetch_array</code> et est presque aussi rapide que <code>pg_fetch_row</code> (la différence est insignifiante).

Note
<code>row</code> est devenu optionnel depuis PHP 4.1.0.

Note

La valeur par défaut de `result_type` a changé pour passer de `PGSQL_BOTH` à `PGSQL_ASSOC` depuis PHP 4.3.0, depuis que les index numériques sont devenus illégaux.

Note

`class_name` et `params` ont été ajoutés dans PHP 5.0. L'ancienne syntaxe avec `result_type` existe toujours pour des compatibilités antérieures.

8.103.31 `pg_fetch_result()` : Retourne les valeurs d'un résultat

string `pg_fetch_result` (resource `result` , int `row` , mixed `field`) string `pg_fetch_result` (resource `result` , mixed `field`)

`pg_fetch_result` retourne la valeur d'une ligne et d'un champ (colonne) en particulier à partir d'un ressource de résultat PostgreSQL.

8.103.32 `pg_fetch_row()` : Lit une ligne dans un tableau

array `pg_fetch_row` (resource `result` , int `row`)

`pg_fetch_row` lit une ligne dans le résultat associé à l'index `result` .

Note

Cette fonction définit les champs NULL à la valeur PHP NULL .

Note

`row` est devenu optionnel depuis PHP 4.1.0.

8.103.33 `pg_field_is_null()` : Teste si un champ PostgreSQL est à NULL

int `pg_field_is_null` (resource `result` , int `row` , mixed `field`) int `pg_field_is_null` (resource `result` , mixed `field`)

`pg_field_is_null` teste si un champ dans une ressource de résultat est NULL .

Note

Auparavant, cette fonction s'appelait `pg_fieldisnull` .

8.103.34 `pg_field_name()` : Retourne le nom d'un champ PostgreSQL

string `pg_field_name` (resource `result` , int `field_number`)

`pg_field_name` retourne le nom du champ qui occupe la colonne numéro `field_number` dans le résultat `result` . La numérotation des champs commence à 0.

Note

Auparavant, cette fonction s'appelait `pg_fieldname` .

8.103.35 `pg_field_num()` : Retourne le numéro d'une colonne

int `pg_field_num` (resource `result` , string `field_name`)

`pg_field_num` retourne le numéro de la colonne, dont le nom est `field_name` , dans le résultat `result` .

Note

Auparavant, cette fonction s'appelait <code>pg_fieldnum</code> .
--

8.103.36 `pg_field prtlen()` : Retourne la taille imprimée

int `pg_field prtlen` (resource `result` , int `row_number` , mixed `field_name_or_number`) int
`pg_field prtlen` (resource `result` , mixed `field_name_or_number`)

`pg_field prtlen` retourne la taille imprimée (nombre de caractères) d'une valeur donnée dans un résultat PostgreSQL. La numérotation des lignes commence à 0. `pg_field prtlen` retourne -1 en cas d'erreur.

Le paramètre `field_name_or_number` peut être passé soit en tant qu' entier , soit en tant que chaîne de caractères . S'il est passé en tant qu' entier , PHP l'identifie comme le numéro d'un champ, sinon, comme le nom d'un champ.

Voir l'exemple donné à la page de la documentation de la fonction `pg_field_name` .

Note

Auparavant, cette fonction s'appelait <code>pg_fieldprtlen</code> .

8.103.37 `pg_field_size()` : Retourne la taille interne de stockage d'un champ donné

int `pg_field_size` (resource `result` , int `field_number`)

`pg_field_size` retourne la taille interne de stockage d'un champ donné, en octets.

Note

Auparavant, cette fonction s'appelait <code>pg_fieldsize</code> .

8.103.38 `pg_field_type_oid()` : Retourne le type ID (OID) pour le numéro du champ correspondant

int `pg_field_type_oid` (resource `result` , int `field_number`)

`pg_field_type_oid` retourne un entier contenant le OID du type de base du champ `field_number` donné dans la ressource PostgreSQL `result` .

Vous pouvez obtenir plus d'informations à propos du type de champ en interrogeant la table système de PostgreSQL `pg_type` avec le OID obtenu par cette fonction.

Note

Si le champ utilise un domaine PostgreSQL (plutôt que d'un type basic), c'est le OID du domaine sous-jacent qui est retourné, plutôt que le OID du domaine en tant que tel.

8.103.39 **pg_field_type()** : Retourne le type d'un champ PostgreSQL donné par index

string **pg_field_type** (resource result , int field_number)

pg_field_type retourne une chaîne contenant le type de base du champ donné par son index `field_number` .

Note

Si le champ utilise un domaine PostgreSQL (plutôt que d'un type basic), c'est le nom du domaine sous-jacent qui est retourné, plutôt que le nom du domaine en tant que tel.

Note

Auparavant, cette fonction s'appelait <u>pg_fieldtype</u> .

8.103.40 **pg_free_result()** : Libère la mémoire

bool **pg_free_result** (resource result)

pg_free_result libère la mémoire et les données associées avec le jeu de résultats PostgreSQL associé.

pg_free_result n'est vraiment utile que si vous risquez d'utiliser trop de mémoire durant votre script. La mémoire occupée par les résultats est automatiquement libérée à la fin du script.

Note

Auparavant, cette fonction s'appelait <u>pg_freeresult</u> .
--

8.103.41 **pg_get_notify()** : Lit le message SQL NOTIFY

array **pg_get_notify** (resource connection , int result_type)

pg_get_notify reçoit le message de NOTIFY envoyé par une commande SQL NOTIFY . Pour lire le message associé, utilisez la commande LISTEN .

8.103.42 **pg_get_pid()** : Lit l'identifiant de processus du serveur PostgreSQL

int **pg_get_pid** (resource connection)

pg_get_pid lit l'identifiant de processus du serveur PostgreSQL. L'identifiant de processus est pratique pour vérifier si un message de NOTIFY a été envoyé via pg_get_notify par un autre processus ou pas.

8.103.43 **pg_get_result()** : Lit un résultat PostgreSQL asynchrone

resource **pg_get_result** (resource connection)

pg_get_result lit la ressource de résultat d'une requête asynchrone exécutée par pg_send_query , pg_send_query_params ou pg_send_execute .

pg_send_query et les autres fonctions de requête asynchrone peuvent envoyer plusieurs requêtes à un serveur PostgreSQL et pg_get_result est utilisé pour obtenir chaque résultat de requête, un par un.

8.103.44 pg_host() : Retourne le nom d'hôte

string pg_host (*resource connection*)

pg_host retourne le nom d'hôte associé à l'index de connexion PostgreSQL.

8.103.45 pg_insert() : Insère un tableau dans une table

mixed pg_insert (*resource connection* , string *table_name* , array *assoc_array* , *int options*)

pg_insert insère le tableau *assoc_array* dans la table *table_name* . Si *options* est spécifiée, pg_convert s'applique à *assoc_array* avec l'option spécifiée.

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

8.103.46 pg_last_error() : Lit le dernier message d'erreur sur la connexion

string pg_last_error (*resource connection*)

pg_last_error retourne le dernier message d'erreur pour une connexion *connection* .

Les messages d'erreur peuvent être écrasés par des appels internes à l'extension PostgreSQL (libpq) : il se peut que le message retourné ne soit pas approprié, notamment si plusieurs erreurs ont eu lieu dans le module.

Utilisez pg_result_error , pg_result_error_field , pg_result_status et pg_connection_status pour améliorer la gestion des erreurs.

Note

Auparavant, cette fonction s'appelait <u>pg_errormessage</u> .
--

8.103.47 pg_last_notice() : Retourne la dernière note du serveur PostgreSQL

string pg_last_notice (*resource connection*)

pg_last_notice retourne la dernière note du serveur PostgreSQL sur la connexion *connection* spécifiée. Le serveur PostgreSQL envoie des notes dans plusieurs cas, par exemple lors de création d'une colonne SERIAL dans une table.

Avec pg_last_notice , vous pouvez éviter des requêtes inutiles en vérifiant si des notes sont liées ou pas à votre transaction.

Attention

Cette fonction est EXPERIMENTALE et elle n'est pas complètement implémentée. `pg_last_notice` a été ajoutée en PHP 4.0.6. Cependant, PHP 4.0.6 avait un problème avec la gestion des notes. L'utilisation du module PostgreSQL avec PHP 4.0.6 n'est pas recommandé, même si vous n'utilisez pas `pg_last_notice`.

Cette fonction est totalement implémentée en PHP 4.3.0. Les versions plus anciennes de PHP ignorent les données de connexion au serveur.

Le suivi des notes peut être rendu optionnel en mettant à 1 la directive de configuration `pgsql.ignore_notice` du fichier `php.ini`, depuis PHP 4.3.0.

L'enregistrement des notes peut être rendu optionnel en mettant la directive de configuration `pgsql.log_notice` du `php.ini` à 0, à partir de PHP 4.3.0. A moins que `pgsql.ignore_notice` ne soit à 0, les notes ne seront pas enregistrées.

8.103.48 pg_last_oid() : Retourne l'identifiant de la dernière ligne

string `pg_last_oid` (resource result)

`pg_last_oid` sert à récupérer le OID assigné à une ligne insérée.

Le champ OID est devenu optionnel depuis PostgreSQL 7.2 et ne sera plus présent par défaut dans PostgreSQL 8.1. Lorsque le champ OID n'est pas présent dans la table, le programmeur doit utiliser `pg_result_status` pour vérifier si la ligne a été correctement insérée.

Pour obtenir la valeur d'un champ SERIAL dans une ligne insérée, il est nécessaire d'utiliser la fonction CURRVAL de PostgreSQL en nommant la séquence à qui la dernière valeur est requise. Si le nom de la séquence est inconnu, la fonction PostgreSQL 8.0 `pg_get_serial_sequence` est nécessaire.

PostgreSQL 8.1 a une fonction LASTVAL qui retourne la valeur de la séquence la plus récemment utilisée de la session. Ceci permet d'éviter de nommer la séquence, la table ou la colonne.

Note

Auparavant, cette fonction s'appelait `pg_getlastoid`.

8.103.49 pg_lo_close() : Ferme un objet de grande taille PostgreSQL

bool `pg_lo_close` (resource large_object)

`pg_lo_close` ferme un objet de type Inversion Large Object. `large_object` est un descripteur de fichier, obtenu avec `pg_lo_open`.

Pour utiliser un objet de grande taille (lo), il est nécessaire de le faire dans une transaction.

Note

Auparavant, cette fonction s'appelait `pg_loclose`.

8.103.50 pg_lo_create() : Crée un objet de grande taille PostgreSQL

int `pg_lo_create` (resource connection)

`pg_lo_create` crée un objet de type Inversion Large Object et retourne son Oid. Les modes d'accès PostgreSQL `INV_READ` , `INV_WRITE` et `INV_ARCHIVE` ne sont pas supportés : l'objet peut toujours être créé avec des droits d'accès en lecture et écriture. Le mode `INV_ARCHIVE` a été supprimé des bases PostgreSQL (version 6.3 et ultérieur).

Pour utiliser un objet de grande taille (`lo`), il est nécessaire de le faire dans une transaction.

Au lieu d'utiliser l'interface d'objet de grande taille ((qui n'a aucun contrôle d'accès et qui est encombrant à utiliser), essayez la colonne de type `bytea` de PostgreSQL et `pg_escape_bytea` .

Note

Auparavant, cette fonction s'appelait <code>pg_locreate</code> .
--

8.103.51 `pg_lo_export()` : Exporte un objet de grande taille vers un fichier

`bool pg_lo_export (resource connection , int oid , string pathname)` `bool pg_lo_export (int oid , string pathname)`

`pg_lo_export` prend un objet de grande taille de la base de données PostgreSQL et sauvegarde son contenu dans un fichier local au système.

Pour utiliser un objet de grande taille (`lo`), il est nécessaire de le faire dans une transaction.

Note

Auparavant, cette fonction s'appelait <code>pg_loexport</code> .
--

8.103.52 `pg_lo_import()` : Importe un objet de grande taille depuis un fichier

`int pg_lo_import (resource connection , string pathname)` `int pg_lo_import (string pathname)`

Dans les versions antérieures à PHP 4.2.0, la syntaxe de cette fonction était différente. Voici la définition :

`int pg_lo_import (string pathname , resource connection)`

`pg_lo_import` crée un nouvel objet de grande taille dans la base de données en utilisant un fichier dans le système de fichier en tant que données sources.

Pour utiliser un objet de grande taille (`lo`), il est nécessaire de le faire dans une transaction.

Note

Lorsque le <code>safe-mode</code> est activé, PHP vérifie si le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.

Note

Auparavant, cette fonction s'appelait <code>pg_loimport</code> .
--

8.103.53 pg_lo_open() : Ouvre un objet de grande taille PostgreSQL

resource **pg_lo_open** (resource connection , int oid , string mode)

pg_lo_open ouvre un objet de type Inversion Large Object et retourne un descripteur de fichier pour cet objet permettant ainsi d'être manipulé.

Attention

Ne fermer pas la connexion à la base de données avant de fermer la ressource de l'objet de grande taille.

Pour utiliser un objet de grande taille (lo), il est nécessaire de le faire dans une transaction.

Note

Auparavant, cette fonction s'appelait pg_loopen .

8.103.54 pg_lo_read_all() : Lit un objet de grande taille en totalité

int **pg_lo_read_all** (resource large_object)

pg_lo_read_all lit un objet de grande taille en totalité et le passe directement au client, après les en-têtes adéquates. Cette fonction est prévue pour transmettre des sons ou des images.

Pour utiliser un objet de grande taille (lo), il est nécessaire de le faire dans une transaction.

Note

Auparavant, cette fonction s'appelait pg_loreadall .

8.103.55 pg_lo_read() : Lit un objet de grande taille

string **pg_lo_read** (resource large_object , int len)

pg_lo_read lit au plus len octets d'un objet de grande taille, et retourne les données sous la forme d'une chaîne.

Pour utiliser un objet de grande taille (lo), il est nécessaire de le faire dans une transaction.

Note

Auparavant, cette fonction s'appelait pg_loread .

8.103.56 pg_lo_seek() : Modifie la position dans un objet de grande taille

bool **pg_lo_seek** (resource large_object , int offset , int whence)

pg_lo_seek modifie la position du pointeur dans l'objet de grande taille large_object .

Pour utiliser un objet de grande taille (lo), il est nécessaire de le faire dans une transaction.

8.103.57 `pg_lo_tell()` : Retourne la position courante dans un objet de grande taille PostgreSQL

int `pg_lo_tell` (resource large_object)

`pg_lo_tell` retourne la position courante (à partir du début) du pointeur de lecture sur l'objet de grande taille `large_object` .

8.103.58 `pg_lo_unlink()` : Efface un objet de grande taille PostgreSQL

bool `pg_lo_unlink` (resource connection , int oid)

`pg_lo_unlink` efface l'objet de grande taille dont l'identifiant est `oid` , pour la connexion `connection` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Pour utiliser un objet de grande taille (`lo`), il est nécessaire de le faire dans une transaction.

Note
Auparavant, cette fonction s'appelait <code>pg_lounlink</code> .

8.103.59 `pg_lo_write()` : Ecrit un objet de grande taille PostgreSQL

int `pg_lo_write` (resource large_object , string data , int len)

`pg_lo_write` écrit des données à l'intérieur d'un objet de grande taille à la position courante.

Pour manipuler un objet de grande taille (`lo`), il est nécessaire de placer les opérations dans un bloc de transaction.

Note
Auparavant, cette fonction s'appelait <code>pg_lowrite</code> .

8.103.60 `pg_meta_data()` : Lit les métadonnées de la table PostgreSQL

array `pg_meta_data` (resource connection , string table_name)

`pg_meta_data` retourne la définition de la table `table_name` sous forme de tableau.

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

8.103.61 `pg_num_fields()` : Retourne le nombre de champ

int `pg_num_fields` (resource result)

`pg_num_fields` retourne le nombre de champs (ou colonnes) d'un résultat PostgreSQL.

Note

Auparavant, cette fonction s'appelait <code>pg_numfields</code> .

8.103.62 `pg_num_rows()` : Retourne le nombre de lignes PostgreSQL

int `pg_num_rows` (resource result)

`pg_num_rows` retourne le nombre de lignes d'un résultat PostgreSQL.

Note

Auparavant, cette fonction s'appelait <code>pg_numrows</code> .

8.103.63 `pg_options()` : Retourne les options PostgreSQL

string `pg_options` (resource connection)

`pg_options` retourne une chaîne contenant les options de la connexion PostgreSQL connection .

8.103.64 `pg_parameter_status()` : Consulte un paramètre de configuration courant du serveur

string `pg_parameter_status` (resource connection , string param_name) string
`pg_parameter_status` (string param_name)

Consulte un paramètre de configuration courant du serveur.

Certaines valeurs de paramètres sont retournées par le serveur automatiquement lors du démarrage de la connexion ou lorsqu'une valeur change. `pg_parameter_status` peut être utilisée pour interroger ces configurations. La fonction retourne la valeur courante du paramètre s'il est connu ou FALSE si le paramètre est inconnu.

Les paramètres retournés par PostgreSQL 8.0 sont `server_version` , `server_encoding` , `client_encoding` , `is_superuser` , `session_authorization` , `DateStyle` , `TimeZone` et `integer_datetimes` . (`server_encoding` , `TimeZone` et `integer_datetimes` n'étaient pas retournés dans les versions antérieures à 8.0.) Notez que `server_version` , `server_encoding` et `integer_datetimes` ne peuvent changer après le démarrage de PostgreSQL.

Les serveurs PostgreSQL 7.3 ou de versions inférieures ne retournent pas de paramètres de configurations, `pg_parameter_status` inclue une logique pour obtenir des valeurs pour `server_version` et `client_encoding` quand même. Les applications devraient utiliser `pg_parameter_status` au lieu du code ad hoc pour déterminer ces valeurs.

Attention

Sur les versions des serveurs PostgreSQL pre-7.4, le changement de <code>client_encoding</code> avec SET après le démarrage de la connexion ne sera pas réfléchi par <code>pg_parameter_status</code> .

8.103.65 `pg_pconnect()` : Établit une connexion PostgreSQL persistante

resource `pg_pconnect` (string connection_string , int connect_type)

`pg_pconnect` retourne une ressource de connexion persistante en cas de succès, ou FALSE en cas d'erreur.

Si un second appel est fait à `pg_pconnect` avec le même `connection_string` comme étant une connexion existante, la connexion existante sera retournée à moins que vous ne passez `PGSQL_CONNECT_FORCE_NEW` à `connect_type` .

Pour activer les connexions persistantes, la directive de configuration `pgsql.allow_persistent` du `php.ini` doit être mise à "On" (ce qui est sa valeur par défaut). Le nombre maximal de connexions peut être limité grâce à la directive de configuration `pgsql.max_persistent` dans le fichier `php.ini` (par défaut, elle vaut -1, c'est à dire pas de limite). Le nombre total de connexions peut être configuré avec la directive `pgsql.max_links` du fichier `php.ini` .

`pg_close` ne refermera pas les connexions persistantes générées par `pg_pconnect` .

8.103.66 `pg_ping()` : Ping la connexion à la base

`bool pg_ping (resource connection)`

`pg_ping` ping la connexion à la base de données et essaie de se reconnecter si la connexion est perdue.

8.103.67 `pg_port()` : Retourne le numéro de port

`int pg_port (resource connection)`

`pg_port` retourne le numéro de port de la connexion identifiée `connection` .

8.103.68 `pg_prepare()` : Envoie une requête pour créer une requête préparée avec les paramètres donnés et attend l'exécution

`resource pg_prepare (resource connection , string stmtname , string query) resource pg_prepare (string stmtname , string query)`

`pg_prepare` crée une requête préparée pour une exécution ultérieure avec `pg_execute` ou `pg_send_execute` . Cette caractéristique permet aux commandes qui seront utilisées à plusieurs reprises d'être analysées et planifiées qu'une seule fois, plutôt que d'être exécutées chaque fois. `pg_prepare` est supportée seulement avec les versions PostgreSQL 7.4 ou plus récentes; la commande échouera si vous l'utilisez avec des versions antérieures.

La fonction crée une requête préparée nommée `stmtname` à partir de la chaîne `query` , celle-ci doit contenir qu'une seule commande SQL. `stmtname` peut être vide ("") pour créer une requête qui n'est pas nommée. Dans ce cas, les requêtes qui existaient et qui se trouvaient sans noms sont automatiquement remplacées; autrement, il y aura une erreur si le nom de la requête est déjà défini dans la session courante. Si des paramètres sont utilisés, ils sont référés à \$1, \$2, etc. dans `query` .

Des requêtes préparées à utiliser avec `pg_prepare` peuvent être aussi créées en exécutant la requête SQL `PREPARE` . (Par contre, `pg_prepare` est plus flexible puisqu'elle ne nécessite pas que les types des paramètres soit pré-spécifiés.) De plus, bien qu'il n'y a pas de fonction PHP pour supprimer une requête préparée, la requête SQL `DEALLOCATE` peut être utilisé pour ce motif.

8.103.69 `pg_put_line()` : Envoie une chaîne au serveur PostgreSQL

`bool pg_put_line (string data)` `bool pg_put_line (resource connection , string data)`

`pg_put_line` envoie une chaîne (terminée par `NULL`) au serveur PostgreSQL. Ceci est nécessaire en conjonction avec une commande `COPY FROM` PostgreSQL.

`COPY` est un chargement très rapide des données supporté par PostgreSQL. Les données sont passées sans être analysées et dans une simple transaction.

Une alternative plutôt que d'utiliser la commande brute `pg_put_line` est d'utiliser `pg_copy_from` . C'est une interface bien plus simple.

Note
Notez que l'application doit explicitement ajouter les deux caractères "\." à la fin de la chaîne pour indiquer au serveur qu'elle a finit d'envoyer des données, avant d'appeler <code>pg_end_copy</code> .
Attention
L'utilisation de <code>pg_put_line</code> cause sur la plupart des objets de grande taille à échouer, incluant <code>pg_lo_read</code> et <code>pg_lo_tell</code> . Vous pouvez utiliser <code>pg_copy_from</code> et <code>pg_copy_to</code> à la place.

8.103.70 `pg_query_params()` : Envoie une commande au serveur et attend le résultat, avec les capacités de passer des paramètres séparément de la commande texte SQL

`resource pg_query_params (resource connection , string query , array params)` `resource pg_query_params (string query , array params)`

Envoie une commande au serveur et attend le résultat, avec les capacités de passer des paramètres séparément de la commande texte SQL.

`pg_query_params` est comme `pg_query` , mais offre des fonctionnalités additionnelles : les valeurs des paramètres peuvent être spécifiées séparément de la ligne de commande propre.

`pg_query_params` est supportée seulement avec les versions PostgreSQL 7.4 ou plus récentes; la commande échouera si vous l'utilisez avec des versions antérieures.

Si des paramètres sont utilisés, ils sont référés à `$1` , `$2` , etc. dans `query` . `params` spécifie les valeurs actuelles des paramètres. Une valeur `NULL` dans ce tableau signifie que le paramètre correspondant est `SQL NULL` .

Le principal avantage de `pg_query_params` sur `pg_query` est que les valeurs des paramètres peuvent être séparées de la requête `query` , par conséquent, on évite les échappements de caractères ennuyeux et source d'erreurs. Contrairement à `pg_query` , `pg_query_params` permet seulement une seule commande SQL dans la chaîne donnée. (Il peut y avoir des points-virgules à l'intérieur mais pas plus d'une seule commande.)

8.103.71 `pg_query()` : Exécute une requête PostgreSQL

`resource pg_query (string query)` `resource pg_query (resource connection , string query)`

`pg_query` exécute la requête `query` sur la base de données spécifiée `connection` .

Si une erreur se produit et FALSE est retourné, les détails de l'erreur peuvent être récupérés en utilisant la fonction `pg_last_error` si la connexion est valide.

Note
Bien que connection puisse être omis, il n'est pas recommandé de le faire, car il peut se révéler difficile de retrouver les bogues dans les scripts.
Note
Auparavant, cette fonction s'appelait <code>pg_exec</code> . <code>pg_exec</code> est toujours disponible pour des raisons de compatibilité, mais les utilisateurs sont encouragés à utiliser le nouveau nom.

8.103.72 `pg_result_error_field()` : Retourne un champ individuel d'un rapport d'erreur

string `pg_result_error_field` (resource result , int fieldcode)

`pg_result_error_field` retourne un des champs détaillés de message d'erreur qui sont associés à la ressource result . Cette fonction est disponible seulement sur les serveurs PostgreSQL 7.4 ou supérieurs. Le champ d'erreur est spécifié par fieldcode .

Puisque `pg_query` et `pg_query_params` retourne FALSE si la requête échoue, vous devez utiliser `pg_send_query` et `pg_get_result` pour obtenir le jeu de résultats.

Si vous avez besoin d'obtenir plus d'informations sur l'erreur lors de l'échec des requêtes avec `pg_query` , utilisez `pg_set_error_verbosity` et `pg_last_error` et analyser ensuite le résultat.

8.103.73 `pg_result_error()` : Lit le message d'erreur associé à un résultat

string `pg_result_error` (resource result)

`pg_result_error` retourne le message d'erreur associé au résultat result . Par conséquent, l'utilisateur a des chances d'obtenir un message d'erreur plus approprié que via `pg_last_error` .

La fonction `pg_result_error_field` peut donner bien plus de détails sur les erreurs que `pg_result_error` .

Comme `pg_query` retourne FALSE si la requête échoue, vous devez utiliser `pg_send_query` et `pg_get_result` pour récupérer la ressource de résultat.

8.103.74 `pg_result_seek()` : Modifie la ligne courant dans un résultat

array `pg_result_seek` (resource result , int offset)

`pg_result_seek` choisit la ligne offset comme ligne courante dans le résultat result .

8.103.75 `pg_result_status()` : Lit le statut du résultat

int `pg_result_status` (resource result)

`pg_result_status` retourne le statut du résultat `result` ou la commande d'exécution de PostgreSQL associée au résultat.

8.103.76 `pg_select()` : Effectue une sélection PostgreSQL

mixed `pg_select` (resource `connection` , string `table_name` , array `assoc_array` , int `options`)

`pg_select` sélectionne les enregistrements par `assoc_array` qui est au format `champ=>valeur` . Lorsque la requête réussit, elle retourne un tableau contenant tous les enregistrements et champs qui vérifient la condition spécifiée par `assoc_array` .

Si `options` est spécifiée, `pg_convert` est appliquée à `assoc_array` avec les drapeaux spécifiés.

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

8.103.77 `pg_send_execute()` : Envoie une requête pour exécuter une requête préparée avec des paramètres donnés, sans attendre le(s) résultat(s)

bool `pg_send_execute` (resource `connection` , string `stmtname` , array `params`)

Envoie une requête pour exécuter une requête préparée avec des paramètres donnés, sans attendre le(s) résultat(s).

Cette fonction est similaire à `pg_send_query_params` , mais la commande qui sera exécutée est spécifiée en nommant une requête préalablement préparée, au lieu de donner une chaîne en tant que requête. Les paramètres de la fonctions sont gérés de la même manière que `pg_execute` . Comme `pg_execute` , la fonction ne fonctionnera pas sur les versions antérieures à PostgreSQL 7.4.

8.103.78 `pg_send_prepare()` : Envoie une requête pour créer une requête préparée avec les paramètres donnés, sans attendre la fin de son exécution

bool `pg_send_prepare` (resource `connection` , string `stmtname` , string `query`)

Envoie une requête pour créer une requête préparée avec les paramètres donnés, sans attendre la fin de son exécution.

Cette fonction est la version asynchrone de `pg_prepare` : elle retourne TRUE si elle a été capable de répartir la requête et FALSE si elle n'a pas été capable. Après un appel réussi, appelez `pg_get_result` pour déterminer si le serveur a créé correctement la requête préparée. Les paramètres de la fonctions sont gérés de la même manière que `pg_execute` . Comme `pg_execute` , la fonction ne fonctionnera pas sur les versions antérieures à PostgreSQL 7.4.

8.103.79 **pg_send_query_params()** : Envoie une commande et sépare les paramètres au serveur sans attendre le(s) résultat(s)

bool **pg_send_query_params** (resource connection , string query , array params)

Envoie une commande et sépare les paramètres au serveur sans attendre le(s) résultat(s).

Cette fonction est équivalente à pg_send_query à l'exception que les paramètres de la requête peuvent être spécifiés séparément de la chaîne de requête query . Les paramètres de la fonctions sont gérés de la même manière que pg_execute . Comme pg_execute , la fonction ne fonctionnera pas sur les versions antérieures à PostgreSQL 7.4 et elle n'autorise qu'une seule commande par requête.

8.103.80 **pg_send_query()** : Exécute une requête PostgreSQL asynchrone

bool **pg_send_query** (resource connection , string query) bool **pg_send_query** (string query)

pg_send_query envoie une requête ou des requêtes de manière asynchrone à la connexion connection . Contrairement à pg_query , elle peut envoyer plusieurs requêtes à la fois au serveur PostgreSQL et obtenir les résultats un par un en utilisant pg_get_result .

L'exécution du script n'est pas bloquée durant l'exécution des requêtes. Utilisez pg_connection_busy pour vérifier si la connexion est occupée (c'est-à-dire la requête est en train d'être exécutée). Les requêtes peuvent être annulées avec pg_cancel_query .

Bien que l'on puisse envoyer plusieurs requêtes en même temps, il n'est pas possible d'envoyer plusieurs requêtes sur une connexion occupée. Si une requête est envoyée alors que la connexion est occupée, elle attendra que la requête précédente soit terminée et perdra tous ses résultats.

8.103.81 **pg_set_client_encoding()** : Choisit l'encodage du client PostgreSQL

int **pg_set_client_encoding** (string encoding) int **pg_set_client_encoding** (resource connection , string encoding)

pg_set_client_encoding fixe l'encodage du client. Elle retourne 0 en cas de succès et -1 en cas d'erreur.

PostgreSQL convertira automatiquement les données dans l'encodage de la base de données vers l'encodage du client.

Note

Auparavant, cette fonction s'appelait <u>pg_setclientencoding</u> .

8.103.82 **pg_set_error_verbosity()** : Détermine la le degré des messages retournés par pg_last_error et pg_result_error

int **pg_set_error_verbosity** (resource connection , int verbosity) int **pg_set_error_verbosity** (int verbosity)

Détermine la le degré des messages retournés par pg_last_error et pg_result_error .

pg_set_error_verbosity fixe le degré d'erreur et retourne le paramètre précédant de la connexion. Avec le mode PGSQL_ERRORS_TERSE , les messages retournés incluent la sévérité, le texte primaire et la position seulement; normalement, cela va entrer sur une seule ligne. Le mode par défaut (PGSQL_ERRORS_DEFAULT) produit des messages qui incluent les messages ci-dessus et des détails, des astuces ou les champs en contexte (ces messages peuvent être étendus sur plusieurs lignes). Le mode PGSQL_ERRORS_VERBOSE tous les champs disponibles. Le changement du degré des messages n'affecte pas les messages disponibles qui proviennent des résultats déjà existants, mais seulement les messages des résultats créés par la suite.

8.103.83 pg_trace() : Active le suivi d'une connexion PostgreSQL

bool **pg_trace** (string pathname , string mode , resource connection)

pg_trace active le suivi des communications entre PHP et le serveur PostgreSQL. Cet historique sera enregistré dans un fichier. Pour comprendre ces lignes, il faut être familier avec le protocole de communication interne à PostgreSQL.

Pour ceux qui le ne sont pas, elles peuvent être utiles pour suivre les requêtes et les erreurs : avec la commande `grep '^To backend' trace.log` , vous pourrez voir les requêtes réellement envoyées au serveur PostgreSQL. Pour plus d'informations, référez-vous à la [Documentation PostgreSQL](#) .

8.103.84 pg_transaction_status() : Retourne le statut de la transaction en cours du serveur

int **pg_transaction_status** (resource connection)

Retourne le statut de la transaction en cours du serveur.

Attention

pg_transaction_status donnera de mauvais résultats lorsque vous l'utiliserez avec un serveur PostgreSQL 7.3 qui a le paramètre autocommit désactivé. La fonctionnalité de autocommit est dépréciée et n'existe plus dans les versions de serveur plus récentes.

8.103.85 pg_tty() : Retourne le nom de TTY associé à la connexion

string **pg_tty** (resource connection)

pg_tty retourne le nom de TTY de la connexion associée à connection .

Note

pg_tty est obsolète, depuis le serveur ne fait plus attention à la configuration TTY, mais demeure pour des raisons de compatibilités.

8.103.86 `pg_unescape_bytea()` : Supprime le protection d'une chaîne de type `bytea`

string `pg_unescape_bytea` (string data)

`pg_unescape_bytea` supprime la protection des caractères de type `bytea`. Elle retourne la chaîne de caractères protégée, pouvant contenir des données binaires.

Note

Lorsque vous utilisez une commande SELECT avec des données de type `bytea`, PostgreSQL retourne des valeurs octales, préfixées avec des anti-slash \ (e.g. \032). Les utilisateurs doivent effectuer la conversion en format binaire eux-mêmes.

`pg_escape_bytea` requiert PostgreSQL 7.2 ou plus récent. Avec PostgreSQL 7.2.0 et 7.2.1, les données de type `bytea` doivent être transtypée lorsque vous activez le support des chaînes de caractères multi-octets. i.e. INSERT INTO test_table (image) VALUES ('\$image_escaped':bytea); . PostgreSQL 7.2.2 ou plus récent ne requiert pas cette manipulation. Toutefois, si le client et le serveur n'utilisent pas le même jeu de caractères, il peut arriver des erreurs. Il faut alors forcer le transtypage manuellement pour éviter cette erreur.

8.103.87 `pg_untrace()` : Termine le suivi d'une connexion PostgreSQL

bool `pg_untrace` (resource connection)

`pg_untrace` termine le suivi d'une connexion PostgreSQL, initiée avec `pg_trace` .

8.103.88 `pg_update()` : Modifie les lignes d'une table

mixed `pg_update` (resource connection , string table_name , array data , array condition , int options)

`pg_update` modifie les lignes de la table `table_name` , qui vérifient la condition `condition` , et leur donne la valeur de `data` . Si `options` est spécifié, `pg_convert` est appliqué à `data` avec les options spécifiées.

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

8.103.89 `pg_version()` : Retourne un tableau avec les versions du client, du protocole et du serveur (si disponible)

array `pg_version` (resource connection)

`pg_version` retourne un tableau avec les versions du client, du protocole et du serveur. Les versions du protocole et du serveur ne sont disponibles que si PHP a été compilé avec PostgreSQL 7.4 ou supérieur.

Pour plus d'informations sur le serveur, utilisez `pg_parameter_status` .

8.104 Fonctions POSIX

8.104.1 Introduction

Ce module contient une interface avec les documents au standard IEEE 1003.1 (POSIX.1), qui ne sont pas accessibles autrement. Par exemple, POSIX.1 définit les fonctions `open()`, `read()`, `write()` et `close()`, qui ont été traditionnellement des fonctions de PHP 3. Certaines fonctionnalités spécifiques ne sont pas encore disponibles, bien que ce module tâche de remédier à cette situation.

Attention
Des données sensibles peuvent être lues en utilisant les fonctions POSIX, comme <code>posix_getpwnam</code> et compagnie. Aucune des fonctions POSIX n'est surveillée lorsque le <code>safe mode</code> est activé. Il est donc fortement recommandé de désactiver les fonctions POSIX (à l'aide de l'option de configuration <code>--disable-posix</code>) si vous travaillez dans un tel environnement.
Note
Cette extension n'est pas disponible sur les plates-formes Windows.

8.104.2 Installation

Le support des fonctions POSIX est activé par défaut. Vous pouvez désactiver ces fonctions en compilant PHP avec l'option `--disable-posix` .

8.104.3 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

- POSIX_F_OK (entier)
Vérifie si le fichier existe.
- POSIX_R_OK (entier)
Vérifie si le fichier existe et a les permissions de lecture.
- POSIX_W_OK (entier)
Vérifie si le fichier existe et a les permissions d'écriture.
- POSIX_X_OK (entier)
Vérifie si le fichier existe et a les permissions d'exécution.
- POSIX_S_IFBLK (entier)
Blokue les fichiers spéciaux
- POSIX_S_IFCHR (entier)
Caractère des fichiers spéciaux
- POSIX_S_IFIFO (entier)
FIFO (pipe nommé) des fichiers spéciaux
- POSIX_S_IFREG (entier)
Fichier normal Normal file
- POSIX_S_IFSOCK (entier)
Socket

Note
Ces constantes sont disponibles depuis PHP 5.1.0. Notez également que quelques unes peuvent ne pas être disponibles sur votre système.

8.104.4 Voir aussi

La section sur le contrôle des processus peut aussi vous intéresser.

- [Introduction](#)
- [Installation](#)
- [Constantes pré-définies](#)
- [Voir aussi](#)
- [posix_access](#)
- [posix_ctermid](#)
- [posix_get_last_error](#)
- [posix_getcwd](#)
- [posix_getegid](#)
- [posix_geteuid](#)
- [posix_getgid](#)
- [posix_getgrgid](#)
- [posix_getgrnam](#)
- [posix_getgroups](#)
- [posix_getlogin](#)
- [posix_getpgid](#)
- [posix_getpgrp](#)
- [posix_getpid](#)
- [posix_getppid](#)
- [posix_getpwnam](#)
- [posix_getpwuid](#)
- [posix_getrlimit](#)
- [posix_getsid](#)
- [posix_getuid](#)
- [posix_isatty](#)
- [posix_kill](#)
- [posix_mkfifo](#)
- [posix_mknod](#)
- [posix_setegid](#)
- [posix seteuid](#)
- [posix_setgid](#)
- [posix_setpgid](#)
- [posix_setsid](#)
- [posix_setuid](#)
- [posix_strerror](#)
- [posix_times](#)
- [posix_ttyname](#)
- [posix_uname](#)

8.104.6 posix_ctermid() : Retourne le chemin du terminal

string **posix_ctermid** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.104.7 `posix_get_last_error()` : Lit le dernier numéro d'erreur généré par la dernière fonction POSIX qui a échoué

int `posix_get_last_error` ()

`posix_get_last_error` retourne le numéro d'erreur retourné par la dernière fonction POSIX qui a échoué. Si aucune erreur n'a été générée, 0 est retourné. Si vous le message d'erreur associé au numéro d'erreur, utilisez `posix_strerror` .

Voir aussi `posix_strerror` .

8.104.8 `posix_getcwd()` : Chemin du dossier de travail courant

string `posix_getcwd` (void)

`posix_getcwd` retourne le chemin absolu du dossier de travail du script courant. `posix_getcwd` retourne FALSE en cas d'erreur.

8.104.9 `posix_getegid()` : Retourne l'ID effectif du groupe du processus courant

int `posix_getegid` (void)

`posix_getegid` retourne l'ID effectif du groupe du processus courant. Reportez-vous à `posix_getgrgid` pour transformer cette information en nom de groupe.

8.104.10 `posix_geteuid()` : Retourne l'UID effectif de l'utilisateur du processus courant

int `posix_geteuid` (void)

`posix_geteuid` retourne l'UID effectif de l'utilisateur du processus courant. Reportez-vous à `posix_getpwuid` pour obtenir le nom d'utilisateur.

8.104.11 `posix_getgid()` : Retourne l'UID du groupe du processus courant

int `posix_getgid` (void)

`posix_getgid` retourne l'UID du groupe du processus courant. Reportez-vous à `posix_getgrgid` pour accéder au nom du groupe.

8.104.12 `posix_getgrgid()` : Retourne des informations sur un groupe

array `posix_getgrgid` (int gid)

`posix_getgrgid` retourne un array d'informations sur un groupe, et FALSE en cas d'échec. Si gid n'est pas un nombre, `posix_getgrgid` retourne NULL et une alerte E_WARNING est générée.

Exemple avec `posix_getgrgid`

```
<?php
$groupid = posix_getegid();
$groupinfo = posix_getgrgid($groupid);

print_r($groupinfo);

?>
```

L'exemple ci-dessus va afficher :

```
Array
(
  [name] => toons
  [passwd] => x
  [members] => Array
  (
    [0] => tom
    [1] => jerry
  )
  [gid] => 42
)
```

Note

Depuis PHP 4.2.0, les membres sont retournés sous la forme d'un tableau de membres. Avant cela, la fonction ne retournait que le nombre de membres du groupe, et les membres étaient identifiés par leur index numérique.

Voir aussi `posix_getegid` , `filegroup` , `stat` et `safe_mode_gid` .

8.104.13 `posix_getgrnam()` : Retourne des informations sur un groupe

array `posix_getgrnam` (string name)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.104.14 `posix_getgroups()` : Retourne les identifiants du groupe du processus courant

array `posix_getgroups` (void)

`posix_getgroups` retourne un tableau contenant les identifiants du groupe du processus courant. Reportez-vous à `posix_getgrgid` pour pouvoir utiliser ces id.

8.104.15 `posix_getlogin()` : Retourne le nom de login

string `posix_getlogin` (void)

`posix_getlogin` retourne le nom de login de l'utilisateur qui possède le processus courant. Reportez-vous à `posix_getpwnam` pour obtenir plus d'informations sur cet utilisateur.

8.104.16 `posix_getpgid()` : Retourne l'id du groupe de processus

int `posix_getpgid` (int pid)

`posix_getpgid` retourne l'id du groupe de processus pour le processus pid .

Ceci n'est pas une fonction POSIX, mais elle est répandue sur les systèmes BSD et System V. Si votre système ne supporte pas cette fonction, la fonction PHP retournera toujours FALSE .

8.104.17 `posix_getpgrp()` : Retourne l'identifiant du groupe de processus

int `posix_getpgrp` (void)

`posix_getpgrp` retourne l'identifiant du groupe de processus du processus courant. Reportez-vous à POSIX.1 et à `getpgrp(2)` dans le manuel de votre système POSIX pour plus d'informations sur les groupes de processus.

8.104.18 `posix_getpid()` : Retourne l'identifiant du processus courant

int `posix_getpid` (void)

`posix_getpid` retourne l'identifiant du processus courant.

8.104.19 `posix_getppid()` : Retourne l'identifiant du processus parent

int `posix_getppid` (void)

`posix_getppid` retourne l'identifiant du processus parent du processus courant.

8.104.20 `posix_getpwnam()` : Retourne des informations sur un utilisateur

array `posix_getpwnam` (string username)

`posix_getpwnam` retourne un tableau associatif qui contient des informations à propos d'un utilisateur, identifié par son nom, passé en paramètre username .

Les éléments du tableau sont :

Elément	Description
name	Le nom contient le nom de l'utilisateur. Généralement, c'est un nom court, de moins de 16 caractères, mais ce n'est pas son nom réel et complet. Cette valeur devrait correspondre au paramètre username et, donc, il est redondant.
passwd	Contient le mot de passe de l'utilisateur, encrypté. Souvent, dans les systèmes utilisant les mots de passe "fantômes", un astérisque est retourné.
uid	L'UID de l'utilisateur.
gid	L'ID du groupe de l'utilisateur. Utilisez la fonction <code>posix_getgrgid</code> pour connaître le nom du groupe, et ses membres.
gecos	GECOS est un terme obsolète qui fait référence aux données de finger, sur un système Honeywell. Le champ, cependant, a survécu, et son contenu a été formalisé par POSIX. Le champ contient une liste, séparée par des virgules, qui contient le nom complet de l'utilisateur, son téléphone professionnel, son numéro de téléphone bureau et son numéro de téléphone personnel. Sur la plupart des systèmes, seul le nom est disponible.
dir	Cet élément contient le chemin absolu jusqu'au dossier racine de l'utilisateur.
shell	Cet élément contient le chemin absolu jusqu'au dossier d'exécution du shell de l'utilisateur.

8.104.21 `posix_getpwuid()` : Retourne des informations sur un utilisateur

array `posix_getpwuid` (int uid)

`posix_getpwuid` retourne un tableau associatif contenant des informations sur un utilisateur repéré par son UID, passé dans le paramètre uid .

Les éléments du tableau sont :

Elément	Description
name	Le nom contient le nom de l'utilisateur. Généralement, c'est un nom court, de moins de 16 caractères, mais ce n'est pas son nom réel et complet.
passwd	Contient le mot de passe de l'utilisateur, encrypté. Souvent, dans les systèmes utilisant les mots de passe "fantômes", un astérisque est retourné.
uid	Cette valeur devrait correspondre au paramètre uid et, donc, il est redondant.
gid	L'ID du groupe de l'utilisateur. Utilisez la fonction <code>posix_getgrgid</code> pour connaître le nom du groupe, et ses membres.
gecos	GECOS est un terme obsolète qui fait référence aux données de finger, sur un système Honeywell. Le champ, cependant, a survécu et son contenu a été formalisé par POSIX. Le champ contient une liste, séparée par des virgules, qui contient le nom complet de l'utilisateur, son téléphone professionnel, son numéro de bureau et son numéro de téléphone personnel. Sur la plupart des systèmes, seul le nom est disponible.
dir	Cet élément contient le chemin absolu jusqu'au dossier racine de l'utilisateur.
shell	Cet élément contient le chemin absolu jusqu'au dossier d'exécution du shell de l'utilisateur.

8.104.22 `posix_getrlimit()` : Retourne les limites système

array `posix_getrlimit` (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.104.23 `posix_getsid()` : Retourne le sid du processus

int `posix_getsid` (int pid)

`posix_getsid` retourne le sid du processus pid . Si pid est à 0, le sid retourné sera celui du processus courant.

Ceci n'est pas une fonction POSIX, mais elle est répandue sur les systèmes BSD et System V. Si votre système ne supporte pas cette fonction, la fonction PHP retournera toujours FALSE .

8.104.24 `posix_getuid()` : Retourne l'ID de l'utilisateur du processus courant

int `posix_getuid` (void)

`posix_getuid` retourne l'ID numérique de l'utilisateur du processus courant. Reportez-vous à `posix_getpwuid` pour accéder au nom d'utilisateur.

8.104.25 `posix_isatty()` : Détermine si un pointeur de fichier est un terminal interactif

bool `posix_isatty` (int fd)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.104.26 `posix_kill()` : Envoie un signal à un processus

bool `posix_kill` (int pid , int sig)

`posix_kill` envoie le signal sig au processus pid . `posix_kill` retourne FALSE , s'il n'a pas pu envoyer le signal, et TRUE sinon.

Reportez-vous à la page de manuel de kill(2) de votre système POSIX, qui contient plus de détails sur les identifiants négatifs de processus, les pid spéciaux 0 et -1, et le signal numéro 0.

8.104.27 `posix_mkfifo()` : Crée un fichier FIFO (first in, first out) (un pipe nommé)

bool `posix_mkfifo` (string pathname , int mode)

`posix_mkfifo` crée un fichier spécial FIFO qui existe dans le système de fichiers, et sert de point de communication bidirectionnel pour les processus.

Le second paramètre mode doit être au format octal (i.e. 0644). Les droits donnés à la nouvelle pile FIFO dépendent aussi de la configuration de la fonction `umask` . Les droits créés sont (mode & ~umask) .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Lorsque le [safe-mode](#) est activé, PHP vérifie si le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.

8.104.28 [posix_mknod\(\)](#) : Crée un fichier spécial ou ordinaire (POSIX.1)

bool [posix_mknod](#) (string *pathname* , int *mode* , int *major* , int *minor*)

[posix_mknod](#) crée un fichier spécial ou ordinaire.

8.104.29 [posix_setegid\(\)](#) : Modifie le GID réel du processus courant

bool [posix_setegid](#) (int *gid*)

[posix_setegid](#) modifie l'identifiant de groupe réel du processus courant. C'est une fonction de haut niveau, et vous aurez besoin des droits appropriés (généralement ceux du super utilisateur), sur votre système, pour l'utiliser.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.104.30 [posix_seteuid\(\)](#) : Modifie l'identifiant d'utilisateur réel (UID) du processus courant

bool [posix_seteuid](#) (int *uid*)

[posix_seteuid](#) modifie l'identifiant d'utilisateur réel du processus courant. C'est une fonction de haut niveau, et vous aurez besoin des droits appropriés (généralement ceux du super utilisateur), sur votre système, pour l'utiliser.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [posix_setgid](#) .

8.104.31 [posix_setgid\(\)](#) : Fixe le GID effectif du processus courant

bool [posix_setgid](#) (int *gid*)

[posix_setgid](#) fixe le GID effectif du processus courant. Reportez-vous à [posix_getgrgid](#) pour transformer cette information en nom de groupe. L'ordre approprié est d'abord [posix_setgid](#) , puis [posix_setuid](#) .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.104.32 [posix_setpgid\(\)](#) : Fixe l'identifiant de groupe de processus

bool [posix_setpgid](#) (int *pid* , int *pgid*)

posix_setpgid ajoute le processus pid au groupe d'id pgid . Reportez-vous à POSIX.1 et setsid(2) dans le manuel de votre système POSIX pour plus d'informations sur le contrôle de tâche. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

8.104.33 **posix_setsid()** : Fait du processus courant un chef de session

int **posix_setsid** (void)

posix_setsid fait du processus courant un chef de session. Reportez-vous à POSIX.1 et setsid(2) dans le manuel de votre système POSIX pour plus d'informations sur le contrôle de tâche. posix_setsid retourne un identifiant de session.

8.104.34 **posix_setuid()** : Fixe l'UID effective du processus courant

bool **posix_setuid** (int uid)

posix_setuid fixe l'UID effective de l'utilisateur du processus courant. Vous devez avoir les privilèges nécessaires (traditionnellement ceux du root) sur votre système pour le faire.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi posix_setgid .

8.104.35 **posix_strerror()** : Lit le message d'erreur associé à un numéro d'erreur POSIX

string **posix_strerror** (int errno)

posix_strerror retourne le message d'erreur POSIX associé au numéro d'erreur errno . Si errno vaut 0 , alors la chaîne "Success" est retournée. La fonction posix_get_last_error est généralement utilisée avec cette fonction pour lire le dernier message d'erreur POSIX.

Voir aussi posix_get_last_error .

8.104.36 **posix_times()** : Utilisation des ressources

array **posix_times** (void)

posix_times retourne un tableau avec les informations sur l'utilisation du CPU. Les indices sont :

- ticks - nombre de ticks depuis le dernier démarrage
- utime - temps utilisateur utilisé par le processus courant.
- stime - temps système utilisé par le processus courant.
- cutime - temps utilisateur utilisé par le processus courant et ses enfants.
- cstime - temps système utilisé par le processus courant et ses enfants.

Attention

Cette fonction n'est pas fiable à utiliser, elle peut retourner des valeurs négatives pour des temps élevés.
--

8.104.37 `posix_ttyname()` : Retourne le nom de device du terminal

string `posix_ttyname` (int `fd`)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.104.38 `posix_uname()` : Retourne le nom du système

array `posix_uname` (void)

`posix_uname` retourne un tableau associatif avec des informations sur le système. Les indices du tableau sont :

- `sysname` - nom du système d'exploitation (exemple : Linux)
- `nodename` - nom du système (e.g. valiant)
- `release` - édition du système d'exploitation (e.g. 2.2.10)
- `version` - version du système d'exploitation (e.g. #4 Tue Jul 20 17:01:36 MEST 1999)
- `machine` - architecture système (e.g. i586)
- `domainname` - DNS domainname (e.g. example.com)

`domainname` est une extension GNU et ne fait pas partie de POSIX.1, ce champ n'est donc accessible que sur les systèmes GNU ou lors de l'utilisation de la GNU libc.

Posix impose que vous n'ayez pas d'a priori sur le format des chaînes, c'est-à-dire que vous ne devez pas vous attendre à avoir forcément trois chiffres pour la version, par exemple.

8.105 Impression

8.105.1 Introduction

Ces fonctions ne sont disponibles que sous Windows 9.x, ME, NT4 et 2000. Elles ont été ajoutées en PHP 4.0.4.

8.105.2 Installation

Cette extension [PECL](#) n'est pas distribuée avec PHP.

Les utilisateurs de Windows doivent activer la bibliothèque `php_printer.dll` dans le `php.ini` pour utiliser ces fonctions. Vous pouvez télécharger cette extension PECL séparée sur les différentes pages de publication PECL (sélectionnez le bon entrepôt en fonction de votre version de PHP) : [PECL pour PHP 4.3.x](#) , [PECL pour PHP 5.0.x](#) or [PECL instable](#) .

8.105.3 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable	Historique
<code>printer.default_printer</code>	""	<code>PHP_INI_ALL</code>	

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#) .

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [printer_abort](#)
- [printer_close](#)
- [printer_create_brush](#)
- [printer_create_dc](#)
- [printer_create_font](#)
- [printer_create_pen](#)
- [printer_delete_brush](#)
- [printer_delete_dc](#)
- [printer_delete_font](#)
- [printer_delete_pen](#)
- [printer_draw_bmp](#)
- [printer_draw_chord](#)
- [printer_draw_ellipse](#)
- [printer_draw_line](#)
- [printer_draw_pie](#)
- [printer_draw_rectangle](#)
- [printer_draw_roundrect](#)
- [printer_draw_text](#)
- [printer_end_doc](#)
- [printer_end_page](#)
- [printer_get_option](#)
- [printer_list](#)
- [printer_logical_fontheight](#)

- [printer_open](#)
- [printer_select_brush](#)
- [printer_select_font](#)
- [printer_select_pen](#)
- [printer_set_option](#)
- [printer_start_doc](#)
- [printer_start_page](#)
- [printer_write](#)

8.105.5 [printer_close\(\)](#) : Ferme une connexion à une imprimante

void **printer_close** (resource handle)

[printer_close](#) ferme la connexion avec l'imprimante représentée par la ressource handle .
[printer_close](#) ferme aussi le contexte d'imprimante actif.

handle doit être une ressource d'imprimante valide.

Exemple avec [printer_close](#)

```
<?php
$handle = printer_open();
printer_close($handle);
?>
```

8.105.6 [printer_create_brush\(\)](#) : Crée une nouvelle brosse

resource **printer_create_brush** (int style , string color)

printer_create_brush crée une nouvelle brosse et retourne une ressource de brosse. Une brosse est utilisée pour remplir des surfaces. Pour un exemple, voyez la fonction [printer_select_brush](#) . color doit être une couleur au format RGB hexadécimal, c'est à dire "000000" pour le noir. style doit prendre la valeur de l'une des constantes suivantes :

- **PRINTER_BRUSH_SOLID** : crée une brosse avec une couleur pleine.
- **PRINTER_BRUSH_DIAGONAL** : crée une brosse avec une direction diagonale ascendante, de droite à gauche (/).
- **PRINTER_BRUSH_CROSS** : crée une brosse avec une forme de croix (+).
- **PRINTER_BRUSH_DIAGCROSS** : crée une brosse avec une forme de croix diagonale (x).
- **PRINTER_BRUSH_FDIAGONAL** : crée une brosse avec une direction diagonale ascendante, de gauche à droite (\).
- **PRINTER_BRUSH_HORIZONTAL** : crée une brosse avec une direction horizontale (-).
- **PRINTER_BRUSH_VERTICAL** : crée une brosse avec une direction horizontale (|).
- **PRINTER_BRUSH_CUSTOM** : crée une brosse à partir d'un fichier BMP. Le second paramètre est alors utilisé pour spécifier le chemin jusqu'au fichier BMP, au lieu d'être une couleur.

8.105.7 [printer_create_dc\(\)](#) : Crée un contexte d'imprimante

void **printer_create_dc** (resource handle)

printer_create_dc crée un nouveau contexte d'imprimante pour l'imprimante handle . Un contexte d'imprimante est utilisé pour paramétrer les objets graphiques d'un document. handle doit être une ressource d'imprimante valide.

Exemple avec printer_create_dc

```
<?php
$handle = printer_open();
printer_start_doc($handle);
printer_start_page($handle);

printer_create_dc($handle);
/* Exploitation du contexte */
printer_set_option($handle, PRINTER_TEXT_COLOR, "333333");
printer_draw_text($handle, 1, 1, "text");
printer_delete_dc($handle);

/* Création d'un autre contexte */
printer_create_dc($handle);
printer_set_option($handle, PRINTER_TEXT_COLOR, "000000");
printer_draw_text($handle, 1, 1, "text");
/* Exploitation du contexte */

printer_delete_dc($handle);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>
```

8.105.8 printer_create_font() : Crée une nouvelle police

resource **printer_create_font** (string face , int height , int width , int font_weight , bool italic , bool underline , bool strikethrough , int orientation)

printer_create_font crée une nouvelle ressource de police. Pour un exemple, voir la fonction printer_select_font . face est le nom de la police. height est sa taille et width sa largeur. Le paramètre font_weight spécifie le poids de la police (400 est normal), et peut prendre comme valeur l'une des constantes suivantes :

- PRINTER_FW_THIN : fixe un poids de police mince (100).
- PRINTER_FW_ULTRALIGHT : fixe un poids de police très léger (200).
- PRINTER_FW_LIGHT : fixe un poids de police léger (300).
- PRINTER_FW_NORMAL : fixe un poids de police normal (400).
- PRINTER_FW_MEDIUM : fixe un poids de police moyen (500).
- PRINTER_FW_BOLD : fixe un poids de police gras (700).
- PRINTER_FW_ULTRABOLD : fixe un poids de police très gras (800).
- PRINTER_FW_HEAVY : fixe un poids de police lourd (900).

italic vaut TRUE pour indiquer que la police est en italique, et FALSE sinon.

underline vaut TRUE pour indiquer que la police est soulignée, et FALSE sinon.

strikethrough vaut TRUE pour indiquer que la police est barrée, et FALSE sinon.

orientation spécifie l'orientation. Pour un exemple, voyez la fonction printer_select_font .

8.105.9 `printer_create_pen()` : Crée un nouveau stylo

resource `printer_create_pen` (int style , int width , string color)

`printer_create_pen` crée un nouveau stylo et retourne une ressource. Un stylo est utilisé pour dessiner des lignes et des courbes. Pour un exemple, voyez la fonction `printer_select_pen`. `color` doit être une couleur au format RGB hexadécimal, "000000" pour le noir. `width` spécifie la largeur du stylo. `style` doit être l'une des constantes suivantes :

- `PRINTER_PEN_SOLID` : crée un stylo plein.
- `PRINTER_PEN_DASH` : crée un stylo pointillé avec des tirets.
- `PRINTER_PEN_DOT` : crée un stylo pointillé avec des points.
- `PRINTER_PEN_DASHDOT` : crée un stylo pointillé avec des tirets et des points.
- `PRINTER_PEN_DASHDOTDOT` : crée un stylo pointillé avec des tirets et des doubles points.
- `PRINTER_PEN_INVISIBLE` : crée un stylo invisible.

8.105.10 `printer_delete_brush()` : Efface une brosse

void `printer_delete_brush` (resource handle)

`printer_delete_brush` efface la brosse représentée par sa ressource `handle`. Pour un exemple, voyez la fonction `printer_select_brush`. Retourne `NULL` en cas de succès, ou `FALSE` si une erreur survient. `handle` doit être une ressource de brosse valide.

8.105.11 `printer_delete_dc()` : Efface un contexte d'imprimante

bool `printer_delete_dc` (resource handle)

`printer_delete_dc` efface le contexte représenté par sa ressource `handle`. Pour un exemple, voyez la fonction `printer_create_dc`. Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec. `handle` doit être une ressource de contexte d'imprimante valide.

8.105.12 `printer_delete_font()` : Efface une police

void `printer_delete_font` (resource handle)

`printer_delete_font` efface la police représentée par sa ressource `handle`. Pour un exemple, voyez la fonction `printer_select_font`. Retourne `NULL` en cas de succès, ou `FALSE` si une erreur survient. `handle` doit être une ressource de police valide.

8.105.13 `printer_delete_pen()` : Efface un stylo

void `printer_delete_pen` (resource handle)

`printer_delete_pen` efface le stylo représenté par sa ressource `handle`. Pour un exemple, voyez la fonction `printer_select_pen`. Retourne `NULL` en cas de succès, ou `FALSE` si une erreur survient. `handle` doit être une ressource de stylo valide.

8.105.14 `printer_draw_bmp()` : Dessine un fichier BMP

bool **printer_draw_bmp** (resource handle , string filename , int x , int y , *int width* , int height)

`printer_draw_bmp` dessine simplement l'image BMP filename , à la position x , y . handle doit être une ressource d'imprimante valide.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec `printer_draw_bmp`

```
<?php
$handle = printer_open();
printer_start_doc($handle, "Mon Document");
printer_start_page($handle);

printer_draw_bmp($handle, "c:\\image.bmp", 1, 1);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>
```

8.105.15 `printer_draw_chord()` : Dessine un arc de cercle

void **printer_draw_chord** (resource handle , int rec_x , int rec_y , int rec_x1 , int rec_y1 , int rad_x , int rad_y , int rad_x1 , int rad_y1)

`printer_draw_chord` dessine simplement un arc de cercle. handle doit être une ressource d'imprimante valide.

rec_x est l'abscisse supérieure gauche du rectangle d'encadrement.

rec_y est l'ordonnée supérieure gauche du rectangle d'encadrement.

rec_x1 est l'abscisse inférieure droite du rectangle d'encadrement.

rec_y1 est l'ordonnée inférieure droite du rectangle d'encadrement.

rad_x est l'abscisse du point de début de l'arc.

rad_y est l'ordonnée du point de début de l'arc.

rad_x1 est l'abscisse du point de fin de l'arc.

rad_y1 est l'ordonnée du point de fin de l'arc.

Exemple avec `printer_draw_chord`

```
<?php
$handle = printer_open();
printer_start_doc($handle, "Mon Document");
printer_start_page($handle);

$pen = printer_create_pen(PRINTER_PEN_SOLID, 2, "000000");
printer_select_pen($handle, $pen);
```

```

$brush = printer_create_brush(PRINTER_BRUSH_SOLID, "2222FF");
printer_select_brush($handle, $brush);

printer_draw_chord($handle, 1, 1, 500, 500, 1, 1, 500, 1);

printer_delete_brush($brush);
printer_delete_pen($pen);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>

```

8.105.16 printer_draw_ellipse() : Dessine une ellipse

void **printer_draw_ellipse** (resource handle , int ul_x , int ul_y , int lr_x , int lr_y)

printer_draw_ellipse dessine une ellipse. handle doit être une ressource d'imprimante valide.

ul_x est l'abscisse supérieure gauche de l'ellipse.

ul_y est l'ordonnée supérieure gauche de l'ellipse.

lr_x est l'abscisse inférieure droite de l'ellipse.

lr_y est l'ordonnée inférieure droite de l'ellipse.

Exemple avec printer_draw_ellipse

```

<?php
$handle = printer_open();
printer_start_doc($handle, "Mon Document");
printer_start_page($handle);

$pen = printer_create_pen(PRINTER_PEN_SOLID, 2, "000000");
printer_select_pen($handle, $pen);

$brush = printer_create_brush(PRINTER_BRUSH_SOLID, "2222FF");
printer_select_brush($handle, $brush);

printer_draw_ellipse($handle, 1, 1, 500, 500);

printer_delete_brush($brush);
printer_delete_pen($pen);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>

```

8.105.17 printer_draw_line() : Dessine une ligne

void **printer_draw_line** (resource printer_handle , int from_x , int from_y , int to_x , int to_y)

printer_draw_line dessine une ligne entre les points de coordonnées from_x , from_y et to_x , to_y en utilisant le stylo courant. printer_handle doit être une ressource d'imprimante valide.

Exemple avec printer_draw_line

```

<?php
$handle = printer_open();
printer_start_doc($handle, "Mon Document");
printer_start_page($handle);

$pen = printer_create_pen(PRINTER_PEN_SOLID, 30, "000000");
printer_select_pen($handle, $pen);

printer_draw_line($handle, 1, 10, 1000, 10);
printer_draw_line($handle, 1, 60, 500, 60);

printer_delete_pen($pen);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>

```

8.105.18 `printer_draw_pie()` : Dessine un secteur angulaire

void **printer_draw_pie** (resource handle , int rec_x , int rec_y , int rec_x1 , int rec_y1 , int rad1_x , int rad1_y , int rad2_x , int rad2_y)

`printer_draw_pie` dessine un secteur angulaire. handle doit être une ressource d'imprimante valide.

rec_x est l'abscisse supérieure gauche du rectangle d'encadrement.

rec_y est l'ordonnée supérieure gauche du rectangle d'encadrement.

rec_x1 est l'abscisse inférieure droite du rectangle d'encadrement.

rec_y1 est l'ordonnée inférieure droite du rectangle d'encadrement.

rad_x est l'abscisse du point de début de l'arc.

rad_y est l'ordonnée du point de début de l'arc.

rad_x1 est l'abscisse du point de fin de l'arc.

rad_y1 est l'ordonnée du point de fin de l'arc.

Exemple avec `printer_draw_pie`

```

<?php
$handle = printer_open();
printer_start_doc($handle, "Mon Document");
printer_start_page($handle);

$pen = printer_create_pen(PRINTER_PEN_SOLID, 2, "000000");
printer_select_pen($handle, $pen);

$brush = printer_create_brush(PRINTER_BRUSH_SOLID, "2222FF");
printer_select_brush($handle, $brush);

printer_draw_pie($handle, 1, 1, 500, 500, 1, 1, 500, 1);

printer_delete_brush($brush);
printer_delete_pen($pen);

printer_end_page($handle);

```

```
printer_end_doc($handle);
printer_close($handle);
?>
```

8.105.19 `printer_draw_rectangle()` : Dessine un rectangle

void **printer_draw_rectangle** (resource handle , int ul_x , int ul_y , int lr_x , int lr_y)

printer_draw_rectangle dessine un rectangle.

handle doit être une ressource d'imprimante valide.

ul_x est l'abscisse du coin supérieur gauche du rectangle.

ul_y est l'ordonnée du coin supérieur gauche du rectangle.

lr_x est l'abscisse du coin inférieur droit du rectangle.

lr_y est l'ordonnée du coin inférieur droit du rectangle.

Exemple avec `printer_draw_rectangle`

```
<?php
$handle = printer_open();
printer_start_doc($handle, "Mon Document");
printer_start_page($handle);

$pen = printer_create_pen(PRINTER_PEN_SOLID, 2, "000000");
printer_select_pen($handle, $pen);

$brush = printer_create_brush(PRINTER_BRUSH_SOLID, "2222FF");
printer_select_brush($handle, $brush);

printer_draw_rectangle($handle, 1, 1, 500, 500);

printer_delete_brush($brush);
printer_delete_pen($pen);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>
```

8.105.20 `printer_draw_roundrect()` : Dessine un rectangle, avec des coins arrondis

void **printer_draw_roundrect** (resource handle , int ul_x , int ul_y , int lr_x , int lr_y , int width , int height)

printer_draw_roundrect dessine un rectangle, avec des coins arrondis.

handle doit être une ressource d'imprimante valide.

ul_x est l'abscisse du coin supérieur gauche du rectangle.

ul_y est l'ordonnée du coin supérieur gauche du rectangle.

`lr_x` est l'abscisse du coin inférieur droit du rectangle.

`lr_y` est l'ordonnée du coin inférieur droit du rectangle.

`width` est la largeur de l'ellipse.

`height` est la longueur de l'ellipse.

Exemple avec `printer_draw_roundrect`

```
<?php
$handle = printer_open();
printer_start_doc($handle, "Mon Document");
printer_start_page($handle);

$pen = printer_create_pen(PRINTER_PEN_SOLID, 2, "000000");
printer_select_pen($handle, $pen);

$brush = printer_create_brush(PRINTER_BRUSH_SOLID, "2222FF");
printer_select_brush($handle, $brush);

printer_draw_roundrect($handle, 1, 1, 500, 500, 200, 200);

printer_delete_brush($brush);
printer_delete_pen($pen);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>
```

8.105.21 `printer_draw_text()` : Dessine un texte

`void printer_draw_text (resource printer_handle , string text , int x , int y)`

`printer_draw_text` dessine le texte `text` à la position `x` , `y` , en utilisant la police de caractères courante. `printer_handle` doit être une ressource d'imprimante valide.

Exemple avec `printer_draw_text`

```
<?php
$handle = printer_open();
printer_start_doc($handle, "Mon Document");
printer_start_page($handle);

$font = printer_create_font("Arial", 72, 48, 400, false, false, false, 0);
printer_select_font($handle, $font);
printer_draw_text($handle, "test", 10, 10);
printer_delete_font($font);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>
```

8.105.22 `printer_end_doc()` : Ferme le document

`bool printer_end_doc (resource handle)`

printer_end_doc ferme le document dans la file d'attente de l'imprimante. Il est maintenant prêt à être imprimé. Pour un exemple, voyez la fonction [printer_start_doc](#) . `handle` doit être une ressource d'imprimante valide.

8.105.23 **printer_end_page()** : Ferme la page active

bool **printer_end_page** (resource `handle`)

[printer_end_page](#) ferme la page active du document en cours. Pour un exemple, voyez la fonction [printer_start_doc](#) . `handle` doit être une ressource d'imprimante valide.

8.105.24 **printer_get_option()** : Lit les informations de configuration de l'imprimante

mixed **printer_get_option** (resource `handle` , string `option`)

[printer_get_option](#) lit la configuration de l'option `option` . `handle` doit être une ressource d'imprimante valide. Lisez la documentation de [printer_set_option](#) pour connaître les valeurs possibles du paramètre `option` . De plus, les valeurs options suivantes sont accessibles en lecture :

- `PRINTER_DEVICENAME` retourne le nom de device de l'imprimante.
- `PRINTER_DRIVERVERSION` retourne la version du pilote de l'imprimante.

Exemple avec [printer_get_option](#)

```
<?php
$handle = printer_open();
echo printer_get_option($handle, PRINTER_DRIVERVERSION);
printer_close($handle);
?>
```

8.105.25 **printer_list()** : Retourne la liste des imprimantes attachées à un serveur

array **printer_list** (int `enumtype` , string `name` , int `level`)

[printer_list](#) liste les imprimantes disponibles et leur fonctionnalités. `level` représente le niveau d'information demandé. Il peut prendre les valeurs de 1, 2, 4 ou 5. `enumtype` doit être l'une des constantes suivantes :

- `PRINTER_ENUM_LOCAL` : liste les imprimantes locales.
- `PRINTER_ENUM_NAME` : liste des imprimantes de `name` , qui peut être un serveur, un domaine ou un fournisseur d'impression.
- `PRINTER_ENUM_SHARED` : ce paramètre ne peut être utilisé seul, il doit être combiné avec d'autres paramètres, grâce à l'opérateur OR. `PRINTER_ENUM_SHARED` détecte les imprimantes localement partagées.
- `PRINTER_ENUM_DEFAULT` : (Win9.x uniquement) liste l'imprimante par défaut.
- `PRINTER_ENUM_CONNECTIONS` : (WinNT/2000 uniquement) liste les imprimantes avec lesquelles l'utilisateur a établi une connexion.
- `PRINTER_ENUM_NETWORK` : (WinNT/2000 uniquement) liste les imprimantes réseau du domaine de l'ordinateur. Valide uniquement si `level` vaut 1.

- `PRINTER_ENUM_REMOTE` : (WinNT/2000 uniquement) liste les imprimantes réseau et les serveurs d'impression du domaine de l'ordinateur. Valide uniquement si level vaut 1.

Exemple avec `printer_list`

```
<?php
/* liste les imprimantes partagées locales */
var_dump(printer_list(PRINTER_ENUM_LOCAL | PRINTER_ENUM_SHARED));
?>
```

8.105.26 `printer_logical_fontheight()` : Lit la taille logique de la police

int `printer_logical_fontheight` (resource handle , int height)

`printer_logical_fontheight` calcule la taille logique de la police pour la taille de height . handle doit être une ressource d'imprimante valide.

Exemple avec `printer_logical_fontheight`

```
<?php
$handle = printer_open();
echo printer_logical_fontheight($handle, 72);
printer_close($handle);
?>
```

8.105.27 `printer_open()` : Ouvre une connexion à une imprimante

resource `printer_open` (string devicename)

`printer_open` tente d'ouvrir une connexion à l'imprimante devicename , puis retourne une ressource en cas de succès, et FALSE sinon.

Si aucun paramètre n'est fourni, `printer_open` tente d'ouvrir une connexion avec l'imprimante par défaut. Si elle n'est pas détectée dans le fichier php.ini avec la directive printer.default_printer , PHP essaie de la détecter.

`printer_open` initialise aussi un contexte d'impression.

Exemple avec `printer_open`

```
<?php
$handle = printer_open("HP Deskjet 930c");
$handle = printer_open();
?>
```

8.105.28 `printer_select_brush()` : Sélectionne une brosse

void `printer_select_brush` (resource printer_handle , resource brush_handle)

`printer_select_brush` sélectionne la brosse brush_handle comme brosse active pour la connexion à l'imprimante printer_handle . Une brosse est utilisée pour remplir des surfaces. Si vous dessinez un rectangle, une brosse sera nécessaire pour dessiner le contenu, et un stylo pour dessiner les bords. Si vous n'avez pas sélectionné de brosse avant de dessiner des formes, elles ne seront pas remplies. printer_handle doit être une ressource d'imprimante valide. brush_handle doit être une ressource de brosse valide.

Exemple avec `printer_select_brush`

```

<?php
$handle = printer_open();
printer_start_doc($handle, "Mon Document");
printer_start_page($handle);

$pen = printer_create_pen(PRINTER_PEN_SOLID, 2, "000000");
printer_select_pen($handle, $pen);
$brush = printer_create_brush(PRINTER_BRUSH_CUSTOM, "c:\\brush.bmp");
printer_select_brush($handle, $brush);

printer_draw_rectangle($handle, 1, 1, 500, 500);

printer_delete_brush($brush);

$brush = printer_create_brush(PRINTER_BRUSH_SOLID, "000000");
printer_select_brush($handle, $brush);
printer_draw_rectangle($handle, 1, 501, 500, 1001);
printer_delete_brush($brush);

printer_delete_pen($pen);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>

```

8.105.29 `printer_select_font()` : Sélectionne une police de caractères

void **printer_select_font** (resource printer_handle , resource font_handle)

`printer_select_font` sélectionne la police font_handle pour écrire le texte sur la connexion à l'imprimante printer_handle . printer_handle doit être une ressource d'imprimante valide. font_handle doit être une ressource de police de caractères valide.

Exemple avec `printer_select_font`

```

<?php
$handle = printer_open();
printer_start_doc($handle, "Mon Document");
printer_start_page($handle);

$font = printer_create_font("Arial", 148, 76, PRINTER_FW_MEDIUM, false, false, false, -50);
printer_select_font($handle, $font);
printer_draw_text($handle, "PHP est tout simplement génial", 40, 40);
printer_delete_font($font);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>

```

8.105.30 `printer_select_pen()` : Sélectionne un stylo

void **printer_select_pen** (resource printer_handle , resource pen_handle)

`printer_select_pen` sélectionne le stylo pen_handle comme stylo actif pour la connexion à l'imprimante printer_handle . Un stylo est utilisé pour dessiner des lignes et des courbes. Autrement dit, si vous dessinez une ligne, le stylo sera utilisé. Si vous dessinez un rectangle, il sera utilisé pour dessiner les contours tandis que la brosse sera utilisée pour remplir le contenu. Si vous n'avez pas

sélectionné de stylo avant de dessiner des formes, elles ne seront pas tracées. `printer_handle` doit être une ressource d'imprimante valide. `pen_handle` doit être une ressource de stylo valide.

Exemple avec `printer_select_pen`

```
<?php
$handle = printer_open();
printer_start_doc($handle, "Mon Document");
printer_start_page($handle);

$pen = printer_create_pen(PRINTER_PEN_SOLID, 30, "2222FF");
printer_select_pen($handle, $pen);

printer_draw_line($handle, 1, 60, 500, 60);

printer_delete_pen($pen);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>
```

8.105.31 `printer_set_option()` : Configure la connexion à l'imprimante

`bool printer_set_option (resource handle , int option , mixed value)`

`printer_set_option` modifie les options suivantes pour la connexion à l'imprimante `handle`. `handle` doit être une ressource d'imprimante valide. `option` peut être l'une des constantes suivantes :

- `PRINTER_COPIES` : indique le nombre de copies à imprimer. valeur doit être un entier.
- `PRINTER_MODE` : spécifie le type de data ("text", "raw" ou "emf"), valeur doit être une chaîne de caractères.
- `PRINTER_TITLE` : spécifie le nom du document, valeur doit être une chaîne de caractères.
- `PRINTER_ORIENTATION` : spécifie l'orientation du papier, valeur peut être `PRINTER_ORIENTATION_PORTRAIT` ou `PRINTER_ORIENTATION_LANDSCAPE`
- `PRINTER_RESOLUTION_Y` : spécifie la résolution en ordonnées, en DPI, valeur doit être un entier.
- `PRINTER_RESOLUTION_X` : spécifie la résolution en abscisse, en DPI, valeur doit être un entier.
- `PRINTER_PAPER_FORMAT` : spécifie un format de papier prédéfini : donnez à valeur la valeur de `PRINTER_FORMAT_CUSTOM` si vous souhaitez utiliser un format de papier personnalisé, grâce aux constantes `PRINTER_PAPER_WIDTH` et `PRINTER_PAPER_LENGTH`. valeur peut alors être l'une des constantes suivantes :
 - ◆ `PRINTER_FORMAT_CUSTOM` : vous laisse spécifier le format de papier.
 - ◆ `PRINTER_FORMAT_LETTER` : spécifie le format standard "letter" (8 1/2 par 11 pouces (2.54cm)).
 - ◆ `PRINTER_FORMAT_LEGAL` : spécifie le format standard "legal" (8 1/2 par 14 pouces (2.54cm)).
 - ◆ `PRINTER_FORMAT_A3` : spécifie le format standard "A3" (297 par 420 millimètres).
 - ◆ `PRINTER_FORMAT_A4` : spécifie le format standard "A4" (210 par 297 millimètres).
 - ◆ `PRINTER_FORMAT_A5` : spécifie le format standard "A5" (148 par 210 millimètres).
 - ◆ `PRINTER_FORMAT_B4` : spécifie le format standard "B4" (250 par 354 millimètres).
 - ◆ `PRINTER_FORMAT_B5` : spécifie le format standard "B5" (182 par 257 millimètres).
 - ◆ `PRINTER_FORMAT_FOLIO` : spécifie le format standard "FOLIO" (8 1/2 par 13 pouces (2.54cm)).
- `PRINTER_PAPER_LENGTH` : si `PRINTER_PAPER_FORMAT` vaut `PRINTER_FORMAT_CUSTOM`, `PRINTER_PAPER_LENGTH` spécifie une longueur

- personnalisée de papier, en millimètres. valeur doit être un entier .
- **PRINTER_PAPER_WIDTH** : si **PRINTER_PAPER_FORMAT** vaut **PRINTER_FORMAT_CUSTOM** , **PRINTER_PAPER_WIDTH** spécifie une largeur personnalisée de papier, en millimètres. valeur doit être un entier .
 - **PRINTER_SCALE** : spécifie le facteur de mise à l'échelle du document. La taille physique de la page imprimée est alors mise à l'échelle avec un facteur égal à valeur /100. Par exemple, si vous donnez un facteur d'échelle de 50, l'impression sera de la moitié de la taille du document original. valeur doit être un entier .
 - **PRINTER_BACKGROUND_COLOR** : spécifie la couleur de fond pour le contexte actuel. valeur doit être une chaîne de caractères contenant une couleur au format RGB hexadécimal : par exemple, "005533".
 - **PRINTER_TEXT_COLOR** : spécifie la couleur du texte pour ce contexte d'imprimante. valeur doit être une chaîne de caractères contenant une couleur au format RGB hexadécimal : par exemple, "005533".
 - **PRINTER_TEXT_ALIGN** : spécifie l'alignement du texte pour le contexte d'imprimante. valeur peut être une combinaison, avec l'opérateur OR, des constantes suivantes :
 - ◆ **PRINTER_TA_BASELINE** : le texte sera aligné sur la ligne de base.
 - ◆ **PRINTER_TA_BOTTOM** : le texte sera aligné sur la ligne de fond.
 - ◆ **PRINTER_TA_TOP** : le texte sera aligné sur la ligne de haut.
 - ◆ **PRINTER_TA_CENTER** : le texte sera centré.
 - ◆ **PRINTER_TA_LEFT** : le texte sera aligné à gauche.
 - ◆ **PRINTER_TA_RIGHT** : le texte sera aligné à droite.

Exemple avec `printer_set_option`

```
<?php
$handle = printer_open();
printer_set_option($handle, PRINTER_SCALE, 75);
printer_set_option($handle, PRINTER_TEXT_ALIGN, PRINTER_TA_LEFT);
printer_close($handle);
?>
```

8.105.32 `printer_start_doc()` : Commence un nouveau document

bool `printer_start_doc` (resource `handle` , string `document`)

`printer_start_doc` crée un nouveau document dans la file d'attente de l'imprimante `handle` . Un document peut contenir plusieurs pages, et il est programmé pour s'imprimer. `handle` doit être une ressource d'imprimante valide. Le paramètre optionnel `document` sert à donner un nom au document.

Exemple avec `printer_start_doc`

```
<?php
$handle = printer_open();
printer_start_doc($handle, "Mon Document");
printer_start_page($handle);

printer_end_page($handle);
printer_end_doc($handle);
printer_close($handle);
?>
```

8.105.33 `printer_start_page()` : Commence une nouvelle page

bool `printer_start_page` (resource handle)

`printer_start_page` crée une nouvelle page pour le document actif. Pour un exemple, voyez la fonction `printer_start_doc`. `handle` doit être une ressource d'imprimante valide.

8.105.34 `printer_write()` : Ecrit des données sur l'imprimante

bool `printer_write` (resource handle , string content)

`printer_write` écrit le contenu de la variable `content` directement sur l'imprimante `handle`. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

`handle` doit être une ressource d'imprimante valide.

Exemple avec `printer_write`

```
<?php
$handle = printer_open();
printer_write($handle, "Texte à imprimer");
printer_close($handle);
?>
```

8.106 Création de document PostScript

8.106.1 Introduction

Ce module permet de créer des documents PostScript. Il a beaucoup de similarités avec l'extension pdf. Actuellement, cet API est pratiquement identique et dans la plupart des cas, seuls les préfixes de chaque fonction sont remplacés de pdf_ par ps_. Cela fonctionne aussi pour les fonctions qui n'ont pas de signification dans le document PostScript (par exemple, l'ajout d'hyperliens) mais aura un effet si le document est converti en PDF.

Les documents créés par cette extension sont parfois même supérieurs aux documents créés avec l'extension pdf, parce que les fonctions de rendement de texte de pslib peuvent gérer le crénage, la coupure de mot et les ligatures qui résultent à un bien meilleur affichage de boîte texte.

8.106.2 Pré-requis

Vous devez au moins PHP 4.3.0 et pslib >= 0.1.12. La bibliothèque ps (pslib) est disponible à <http://pslib.sourceforge.net/> .

8.106.3 Installation

Une courte note pour l'installation : tapez seulement
\$ pear install ps
dans votre console.

8.106.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.106.5 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Les deux tables suivantes listent toutes les constantes définies par l'extension ps.

Nom	Description
ps_LINECAP_BUTT	
ps_LINECAP_ROUND	
ps_LINECAP_SQUARED	
Nom	Description
ps_LINEJOIN_MITER	
ps_LINEJOIN_ROUND	
ps_LINEJOIN_BEVEL	

8.106.6 Contact Information

Si vous avez des commentaires, résolutions de bogues, améliorations pour soit cette extension ou pslib alors laissez moi un message à steinm@php.net . Toute aide est la bienvenue.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Constantes pré-définies](#)
- [Contact Information](#)
- [ps_add_bookmark](#)
- [ps_add_launchlink](#)
- [ps_add_loclink](#)
- [ps_add_note](#)
- [ps_add_pdflink](#)
- [ps_add_weblink](#)
- [ps_arc](#)
- [ps_arcn](#)
- [ps_begin_page](#)
- [ps_begin_pattern](#)
- [ps_begin_template](#)
- [ps_circle](#)
- [ps_clip](#)
- [ps_close_image](#)
- [ps_close](#)
- [ps_closepath_stroke](#)
- [ps_closepath](#)
- [ps_continue_text](#)
- [ps_curveto](#)
- [ps_delete](#)
- [ps_end_page](#)
- [ps_end_pattern](#)
- [ps_end_template](#)
- [ps_fill_stroke](#)
- [ps_fill](#)
- [ps_findfont](#)
- [ps_get_buffer](#)
- [ps_get_parameter](#)
- [ps_get_value](#)
- [ps_hyphenate](#)
- [ps_lineto](#)
- [ps_makespotcolor](#)
- [ps_moveto](#)
- [ps_new](#)
- [ps_open_file](#)
- [ps_open_image_file](#)
- [ps_open_image](#)
- [ps_place_image](#)
- [ps_rect](#)
- [ps_restore](#)
- [ps_rotate](#)
- [ps_save](#)
- [ps_scale](#)

- [ps_set_border_color](#)
- [ps_set_border_dash](#)
- [ps_set_border_style](#)
- [ps_set_info](#)
- [ps_set_parameter](#)
- [ps_set_text_pos](#)
- [ps_set_value](#)
- [ps_setcolor](#)
- [ps_setdash](#)
- [ps_setflat](#)
- [ps_setfont](#)
- [ps_setgray](#)
- [ps_setlinecap](#)
- [ps_setlinejoin](#)
- [ps_setlinewidth](#)
- [ps_setmiterlimit](#)
- [ps_setpolydash](#)
- [ps_shading_pattern](#)
- [ps_shading](#)
- [ps_shfill](#)
- [ps_show_boxed](#)
- [ps_show_xy](#)
- [ps_show](#)
- [ps_string_geometry](#)
- [ps_stringwidth](#)
- [ps_stroke](#)
- [ps_symbol_name](#)
- [ps_symbol_width](#)
- [ps_symbol](#)
- [ps_translate](#)

8.106.8 **ps_add_launchlink()** : Ajoute un lien qui démarre un fichier

bool **ps_add_launchlink** (resource psdoc , float llx , float lly , float urx , float ury , string filename)

Place un hyperlien à la position donnée pointant vers un programme qui sera démarré lors du clic sur celui-ci. Le lancement de l'hyperlien est un rectangle avec son coin bas gauche à (llx, lly) et son coin haut droit à (urx, ury). Le rectangle a par défaut une mince bordure bleue.

L'hyperlien ne sera pas visible si le document est imprimé ou vu mais sera visible si le document est converti en pdf par soit Acrobat Distiller® ou Ghostview.

8.106.9 **ps_add_loclink()** : Ajoute un lien à une page dans le même document

bool **ps_add_loclink** (resource psdoc , float llx , float lly , float urx , float ury , int page , string dest)

Place un hyperlien à une position donnée pointant à une page dans le même document. Cliquer sur le lien sautera à la page donnée. La première page dans un document a le numéro 1.

Le début de l'hyperlien est un rectangle avec son coin bas gauche à (llx , lly) et son coin haut droit à (urx , lly). Le rectangle a par défaut une mince bordure bleue.

L'hyperlien ne sera pas visible si le document est imprimé ou vu mais sera visible si le document est converti en pdf par soit Acrobat Distiller® ou Ghostview.

8.106.10 ps_add_note() : Ajoute une note à la page courante

bool **ps_add_note** (resource psdoc , float llx , float lly , float urx , float ury , string contents , string title , string icon , int open)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.106.11 ps_add_pdflink() : Ajoute un lien à une page à un second document pdf

bool **ps_add_pdflink** (resource psdoc , float llx , float lly , float urx , float ury , string filename , int page , string dest)

Place un hyperlien à une position donnée à un second document pdf. Cliquer sur le lien branchera au document à la page donnée. La première page dans un document a le numéro 1.

Le début de l'hyperlien est un rectangle avec son coin bas gauche à (llx , lly) et son coin haut droit à (urx , lly). Le rectangle a par défaut une mince bordure bleue.

L'hyperlien ne sera pas visible si le document est imprimé ou vu mais sera visible si le document est converti en pdf par soit Acrobat Distiller® ou Ghostview.

8.106.12 ps_add_weblink() : Ajoute un lien à un emplacement web

bool **ps_add_weblink** (resource psdoc , float llx , float lly , float urx , float ury , string url)

Place un hyperlien à la position donnée à une page web. Le début de l'hyperlien est un rectangle avec son coin bas gauche à (llx, lly) et son coin haut droit à (urx, ury). Le rectangle a par défaut une mince bordure bleue.

L'hyperlien ne sera pas visible si le document est imprimé ou vu mais sera visible si le document est converti en pdf par soit Acrobat Distiller® ou Ghostview.

8.106.13 ps_arc() : Dessine un arc dans le sens contraire des aiguilles d'une montre

bool **ps_arc** (resource psdoc , float x , float y , float radius , float alpha , float beta)

Dessine une portion d'un cercle avec un point milieu à (x , y). L'arc débute à un angle de alpha et termine avec un angle de beta . Il est tracé dans le sens contraire des aiguilles d'une montre. La sous trajectoire ajoutée à la trajectoire courante débute sur l'arc à l'angle alpha et termine sur l'arc à l'angle beta .

8.106.14 **ps_arcn()** : Dessine un arc dans le sens des aiguilles d'une montre

bool **ps_arcn** (resource psdoc , float x , float y , float radius , float alpha , float beta)

Dessine une portion d'un cercle avec un point milieu à (x , y). L'arc débute à un angle de alpha et termine avec un angle de beta . Il est tracé dans le sens des aiguilles d'une montre. La sous trajectoire ajoutée à la trajectoire courante débute sur l'arc à l'angle beta et termine sur l'arc à l'angle alpha .

8.106.15 **ps_begin_page()** : Démarre une nouvelle page

bool **ps_begin_page** (resource psdoc , float width , float height)

Démarre une nouvelle page. Bien que les paramètres width et height implique une taille différente des feuilles pour chaque page, cela n'est pas possible dans PostScript. Le premier appel de ps_begin_page fixera la taille des pages pour le document entier. Des appels consécutifs n'auront aucun effet, à l'exception pour créer une nouvelle page. La situation est différente si vous tentez de convertir le document PostScript en PDF. Cette fonction place des marques pdf à l'intérieur du document qui peuvent fixer la taille pour chaque page individuellement. Le document PDF résultant aura différentes tailles de pages.

Chaque page est encapsulée en sauvegarde/restauration. Cela signifie, que la plupart des configurations effectuées sur une page ne seront pas retenues sur la page suivante.

S'il n'y a aucun appel de ps_findfond jusqu'au premier appel de ps_begin_page , alors l'en-tête du document PostScript sera affichée et le rectangle englobant sera fixé à la taille de la première page. Si ps_findfont était appelée avant que l'en-tête ne soit déjà affichée, le document n'aura pas un rectangle englobant valide. Afin de prévenir cela, vous devriez appeler ps_set_info pour fixer le champ d'information BoundingBox et possiblement Orientation avant d'appeler ps_findfont ou ps_begin_page .

8.106.16 **ps_begin_pattern()** : Démarre un nouveau motif

bool **ps_begin_pattern** (resource psdoc , float width , float height , float xstep , float ystep , int painttype)

Démarre un nouveau motif.

8.106.17 **ps_begin_template()** : Démarre un nouveau modèle

bool **ps_begin_template** (resource psdoc , float width , float height)

Démarre un nouveau modèle.

8.106.18 **ps_circle()** : Dessine un cercle

bool **ps_circle** (resource psdoc , float x , float y , float radius)

Dessine un cercle avec son point milieu à (x , y). Le cercle débute et termine à la position ($x + \text{radius}$, y). Si cette fonction est appelée à l'extérieur d'un chemin, elle démarrera un nouveau chemin. Si elle est appelée à l'intérieur d'un chemin, elle ajoutera le cercle comme étant un sous chemin. Si la dernière opération de dessin ne se termine pas au point ($x + \text{radius}$, y) alors il y aura un trou dans le chemin.

8.106.19 **ps_clip()** : Attache le dessin au chemin courant

bool **ps_clip** (resource psdoc)

Prend le chemin courant et l'utilise pour définir la bordure du secteur à attacher.

8.106.20 **ps_close_image()** : Ferme l'image et libère la mémoire

void **ps_close_image** (resource psdoc , int imageid)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.106.21 **ps_close()** : Ferme le document PostScript

bool **ps_close** (resource psdoc)

Ferme le document PostScript.

Cette fonction écrit une dernière ligne dans le document PostScript qui est un commentaire PostScript contenant le nombre de pages. Elle écrit aussi l'arbre des signets.

8.106.22 **ps_closepath_stroke()** : Ferme et trace un chemin

bool **ps_closepath_stroke** (resource psdoc)

Connecte le dernier point avec le premier point du chemin et dessine le résultat de la ligne fermée.

8.106.23 **ps_closepath()** : Ferme un chemin

bool **ps_closepath** (resource psdoc)

Connecte le dernier point avec le premier point du chemin.

8.106.24 **ps_continue_text()** : Continue un texte à la ligne suivante

bool **ps_continue_text** (resource psdoc , string text)

Écrit un texte une ligne en dessous de la dernière ligne écrite. L'espace entre les lignes est pris avec la valeur "leading" qui doit être fixée avec [ps_set_value](#) en points. La position actuelle du texte est déterminée par les valeurs "textx" et "texty" qui peuvent être récupérées par [ps_get_value](#) .

8.106.25 **ps_curveto()** : Dessine une courbe

bool **ps_curveto** (resource psdoc , float x1 , float y1 , float x2 , float y2 , float x3 , float y3)

Dessine une courbe décrite par les trois points donnés.

8.106.26 **ps_delete()** : Supprime toutes les ressources d'un document PostScript

bool **ps_delete** (resource psdoc)

Libère principale la mémoire utilisée par le document. Ferme aussi un fichier, s'il n'était pas fermée auparavant avec la fonction **ps_close** . Vous devriez dans tous les cas fermer le fichier avec **ps_close** avant, car **ps_close** ne ferme pas simplement le fichier mais écrit aussi une dernière ligne contenant le commentaire PostScript avec le nombre de pages dans le document. Sans ce commentaire, un visualiseur PostScript pourrait ne pas afficher le nombre de pages correctement.

8.106.27 **ps_end_page()** : Termine une page

bool **ps_end_page** (resource psdoc)

Termine une page qui a été débutée avec **ps_begin_page** .

8.106.28 **ps_end_pattern()** : Termine un motif

bool **ps_end_pattern** (resource psdoc)

Termine un motif qui a été débuté avec **ps_begin_pattern** .

8.106.29 **ps_end_template()** : Termine un modèle

bool **ps_end_template** (resource psdoc)

Termine un modèle qui a été débuté avec **ps_begin_template** .

8.106.30 **ps_fill_stroke()** : Remplit et trace le chemin courant

bool **ps_fill_stroke** (resource psdoc)

Remplit et dessine le chemin construit avec des fonctions de dessin comme **ps_lineto** précédemment appelées.

8.106.31 **ps_fill()** : Remplit le chemin courant

bool **ps_fill** (resource psdoc)

Remplit le chemin construit avec des fonctions de dessin comme ps_lineto précédemment appelées.

8.106.32 ps_findfont() : Charge une police

int **ps_findfont** (resource psdoc , string fontname , string encoding , *bool embed*)

Charge une police pour utilisation ultérieure. Avant que le texte soit écrit avec la police chargée, il doit être fixé avec ps_setfont . Cette fonction doit avoir le fichier de police métrique "adobe" afin de calculer l'espace utilisé par les caractères. Une police qui est chargée dans une page sera seulement disponible sur cette page. Les polices qui seront utilisées dans le document complet doivent être chargées avant le premier appel de ps_begin_page . L'appel de ps_findfont entre les pages mettra la cette police disponible pour toutes les pages qui suivent.

8.106.33 ps_get_buffer() : Récupère le tampon complet contenant les données PS générées

string **ps_get_buffer** (resource psdoc)

Pas encore implémentée

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.106.34 ps_get_parameter() : Récupère certains paramètres

string **ps_get_parameter** (resource psdoc , string name , *float modifier*)

Récupère plusieurs paramètre qui ont été fixés par ps_set_parameter . Les paramètres sont par définitions des valeurs de chaîne de caractères . Cette fonction ne peut être utilisée pour récupérer les ressources qui ont été aussi fixées par ps_set_parameter .

8.106.35 ps_get_value() : Récupère certaines valeurs

float **ps_get_value** (resource psdoc , string name , *float modifier*)

Récupère plusieurs valeurs qui ont été fixées par ps_set_value . Les paramètres sont par définition des valeurs de nombre à virgule flottante .

8.106.36 ps_hyphenate() : Relie un mot

array **ps_hyphenate** (resource psdoc , string text)

Relie le mot passé. ps_hyphenate évalue la valeur hypheminchars et le paramètre hyphendict . hyphendict doit être fixé avant d'appeler cette fonction.

Cette fonction requiert que la configuration locale LC_CTYPE soit correctement faite. Cela est fait lorsque l'extension est initialisée en utilisant les variables d'environnement. Sur les systèmes Unix,

lisez les pages de manuel de locale.

8.106.37 **ps_lineto()** : Dessine une ligne

bool **ps_lineto** (resource psdoc , float x , float y)

Ajoute une ligne droite à partir du point courant aux coordonnées données au chemin courant. Utilisez [ps_moveto](#) pour fixer le point de départ de la ligne.

8.106.38 **ps_makespotcolor()** : Crée une tache de couleur

int **ps_makespotcolor** (resource psdoc , string name , float reserved)

Crée une tache de couleur à partir de la couleur de remplissage courante. La couleur de remplissage doit être définie en espace colorimétrique rgb, cmyk ou gris. Le nom de la tache de couleur peut être un nom arbitraire. Une tache de couleur peut être fixée à n'importe quelle couleur avec [ps_setcolor](#) .

8.106.39 **ps_moveto()** : Fixe le point courant

bool **ps_moveto** (resource psdoc , float x , float y)

Fixe le point courant à de nouvelles coordonnées. Si cela est le premier appel de [ps_moveto](#) après qu'un chemin précédent ait été terminé alors il démarrera un nouveau chemin. Si cette fonction est appelée au milieu d'un chemin, elle fixera simplement le point courant et débutera un sous chemin.

8.106.40 **ps_new()** : Crée un nouvel objet document PostScript

resource **ps_new** ()

Crée une nouvelle instance de document. Elle ne crée pas le fichier sur le disque ou dans la mémoire. Elle prépare simplement tout.

8.106.41 **ps_open_file()** : Ouvre un fichier pour écriture

bool **ps_open_file** (resource psdoc , string filename)

Crée un nouveau fichier sur le disque et écrit le document PostScript à l'intérieur. Le fichier sera fermé lorsque [ps_close](#) sera appelée.

8.106.42 **ps_open_image_file()** : Ouvre une image à partir d'un fichier

int **ps_open_image_file** (resource psdoc , string type , string filename , string stringparam , int intparam)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.106.43 **ps_open_image()** : Lit une image pour placement futur

int **ps_open_image** (resource psdoc , string type , string source , string data , int lenght , int width , int height , int components , int bpc , string params)

Lit une image qui est déjà disponible en mémoire. Le paramètre source n'est présent pas évalué et est assumé d'être memory .

8.106.44 **ps_place_image()** : Place une image sur la page

bool **ps_place_image** (resource psdoc , int imageid , float x , float y , float scale)

Place une image précédemment chargée sur la page. L'image peut être redimensionnée.

8.106.45 **ps_rect()** : Dessine un rectangle

bool **ps_rect** (resource psdoc , float x , float y , float width , float height)

Dessine un rectangle avec son coin bas gauche à (x , y). Le rectangle débute et termine sur son coin bas gauche. Si cette fonction est appelée à l'extérieur d'un chemin, elle démarrera un nouveau chemin. Si elle est appelée à l'intérieur d'un chemin, elle ajoutera le rectangle comme étant un sous chemin. Si la dernière opération de dessin ne se termine pas au coin bas gauche alors il y aura un trou dans le chemin.

8.106.46 **ps_restore()** : Restaure un contexte précédemment sauvegardé

bool **ps_restore** (resource psdoc)

Restaure un contexte graphique précédemment sauvegardé. Tout appel de ps_save doit être accompagné par un appel à ps_restore .

8.106.47 **ps_rotate()** : Fixe le facteur de rotation

bool **ps_rotate** (resource psdoc , float rot)

Fixe la rotation d'un système coordonné.

8.106.48 **ps_save()** : Sauvegarde le contexte courant

bool **ps_save** (resource psdoc)

Sauvegarde le contexte graphique courant, contenant les configurations des couleurs, des translations et rotations et quelques autres. Un contexte sauvegardé peut être restauré avec ps_restore .

8.106.49 ps_scale() : Fixe le facteur de redimensionnement

bool **ps_scale** (resource psdoc , float x , float y)

Fixe le redimensionnement horizontal et vertical d'un système coordonné.

8.106.50 ps_set_border_color() : Fixe la couleur des bordures pour les commentaires

bool **ps_set_border_color** (resource psdoc , float red , float green , float blue)

Les commentaires comme les points de début des liens sont entourés d'un rectangle. Cette fonction fixe la couleur de la ligne de bordure.

8.106.51 ps_set_border_dash() : Fixe la grandeur des tirets pour les bordures des commentaires

bool **ps_set_border_dash** (resource psdoc , float black , float white)

Les commentaires comme les points de début des liens sont entourés d'un rectangle. Cette fonction fixe la portion de noir et blanc de la ligne de bordure pointillée.

8.106.52 ps_set_border_style() : Fixe le style de bordure des commentaires

bool **ps_set_border_style** (resource psdoc , string style , float width)

Les commentaires comme les points de début des liens sont entourés d'un rectangle. Cette fonction fixe l'apparence et la largeur de la ligne de bordure.

8.106.53 ps_set_info() : Fixe les champs d'informations d'un document

bool **ps_set_info** (resource p , string key , string val)

Fixe certains champs d'informations du document. Les champs seront montrés en tant que commentaire dans l'en-tête du fichier PostScript. Les valeurs qui peuvent être fixées sont Keywords , Subject , Title , Creator , Author , BoundingBox et Orientation . Soit conscient que certaines de celles-ci ont une signification aux visualiseurs PostScript.

Le BoundingBox est normalement fixé à la valeur donnée par la première page. Cela fonctionne seulement si [ps_findfont](#) n'a pas été appelée auparavant. Dans ces cas, le BoundingBox sera laissé non fixé à moins que vous le fixiez explicitement avec cette fonction.

8.106.54 ps_set_parameter() : Fixe certains paramètres

bool **ps_set_parameter** (resource psdoc , string name , string value)

Fixe plusieurs paramètres qui sont utilisés par beaucoup de fonctions. Les paramètres sont par définition des valeurs de chaîne de caractères .

8.106.55 **ps_set_text_pos()** : Fixe la position l'écriture du texte

bool **ps_set_text_pos** (resource psdoc , float x , float y)

Fixe la position pour la prochaine écriture de texte. Vous devriez alternativement fixer la valeur x et y séparément en appelant ps_set_value et choisir textx respectivement texty comme nom de valeur.

8.106.56 **ps_set_value()** : Fixe certaines valeurs

bool **ps_set_value** (resource psdoc , string name , float value)

Fixe plusieurs valeurs qui sont utilisées par beaucoup de fonctions. Les paramètres sont par définitions des valeurs de nombre à virgule flottante .

8.106.57 **ps_setcolor()** : Fixe la couleur courante

bool **ps_setcolor** (resource psdoc , string type , string colorspace , float c1 , float c2 , float c3 , float c4)

Fixe la couleur pour le dessin, remplissage ou les deux.

8.106.58 **ps_setdash()** : Fixe l'apparence d'une ligne pointillée

bool **ps_setdash** (resource psdoc , float on , float off)

Fixe la longueur des portions de noir et blanc d'une ligne pointillée.

8.106.59 **ps_setflat()** : Fixe la position à plat

bool **ps_setflat** (resource psdoc , float value)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.106.60 **ps_setfont()** : Fixe la police à utiliser pour la prochaine écriture

bool **ps_setfont** (resource psdoc , int fontid , float size)

Fixe une police, qui doit être chargée avant avec ps_findfont . L'écriture de texte sans configurer auparavant une police provoquera une erreur.

8.106.61 ps_setgray() : Fixe la valeur de gris

bool **ps_setgray** (resource psdoc , float gray)

Fixe la valeur de gris pour toutes les opérations de dessin suivantes.

8.106.62 ps_setlinecap() : Fixe l'apparence des fins de ligne

bool **ps_setlinecap** (resource psdoc , int type)

Fixe comment les fins de ligne ressemblent.

8.106.63 ps_setlinejoin() : Fixe comment les lignes connectés sont jointes

bool **ps_setlinejoin** (resource psdoc , int type)

Fixe comment les lignes sont jointes.

8.106.64 ps_setlinewidth() : Fixe la largeur d'une ligne

bool **ps_setlinewidth** (resource psdoc , float width)

Fixe la largeur d'une ligne pour toutes les opérations de dessin suivantes.

8.106.65 ps_setmiterlimit() : Fixe les limites de l'anglet

bool **ps_setmiterlimit** (resource psdoc , float value)

Si deux lignes se joignent dans un petit angle et le joint des lignes est fixé à l'anglet, alors la mèche résultante sera vraiment longue. La limite de l'anglet est le ratio maximal de la longueur de l'anglet (la longueur de la mèche) et la largeur de la ligne.

8.106.66 ps_setpolydash() : Fixe l'apparence d'une ligne pointillée

bool **ps_setpolydash** (resource psdoc , float arr)

Fixe la longueur des portions de noir et blanc d'une ligne pointillée. `ps_setpolydash` est utilisée pour fixer des motifs pointillés plus compliqués.

8.106.67 ps_shading_pattern() : Crée un motif basé sur le ton

int **ps_shading_pattern** (resource psdoc , int shadingid , string optlist)

Crée un motif basé sur le ton, qui doit avoir été créé avant avec `ps_shading`. Les motifs de ton peuvent être utilisés comme des motifs réguliers.

8.106.68 **ps_shading()** : Crée un ton pour usage futur

int **ps_shading** (resource psdoc , string type , float x0 , float y0 , float x1 , float y1 , float c1 , float c2 , float c3 , float c4 , string optlist)

Crée un ton, qui peut être utilisé par [ps_shfill](#) ou [ps_shading_pattern](#) .

La couleur du ton peut être n'importe quelle couleur d'espace excepté pour pattern .

8.106.69 **ps_shfill()** : Remplit un espace avec un ton

bool **ps_shfill** (resource psdoc , int shadingid)

Remplit un espace avec un ton, qui doit être créé avant avec [ps_shading](#) .

8.106.70 **ps_show_boxed()** : Écriture de texte dans une boîte

int **ps_show_boxed** (resource psdoc , string text , float left , float bottom , float width , float height , string hmode , *string feature*)

Écrit du texte dans une boîte donnée. Le coin bas gauche de la boîte est à (left , bottom). Les coupures de ligne sera insérées où nécessaires. Des espaces multiples sont traités comme une seule. Les tabs sont traités comme des espaces.

Le texte sera relié si le paramètre "hyphenation" est fixé à TRUE et le paramètre "hyphendict" contient un fichier valide pour un fichier de liaison. L'espace des lignes est pris à partir de la valeur "leading". Les paragraphes peuvent être séparés par une ligne vide comme dans TeX. Si la valeur "parindent" est fixée à une valeur > 0.0, alors les premières n lignes seront indentées. Le nombre n de lignes est fixé par le paramètre "numindentlines". Afin de prévenir l'indentation des premiers m paragraphes, fixés la valeur "parindentskip" à un nombre positif.

8.106.71 **ps_show_xy()** : Écrit du texte à la position donnée

bool **ps_show_xy** (resource psdoc , string text , float x , float y)

Écrit du texte à la position de texte donnée.

8.106.72 **ps_show()** : Écrit du texte

bool **ps_show** (resource psdoc , string text)

Écrit du texte à la position courante du texte. Cette fonction émettra une erreur si aucune police a été fixée auparavant avec [ps_setfont](#) .

[ps_show](#) évalue les paramètres et les variables suivantes fixés par [ps_set_parameter](#) et [ps_set_value](#) .

charspacing (valeur)

Distance entre deux glyphes consécutifs. Si cette valeur n'est pas égal à zéro, alors toutes les ligatures seront résolues. Les valeurs inférieures à zéro sont autorisées.

Kerning (paramètre)

Fixer ce paramètre à FALSE désactivera le crénage. Le crénage est activé par défaut.

Ligatures (paramètre)

Fixer ce paramètre à FALSE désactivera l'utilisation des ligatures. Les ligatures sont activées par défaut.

Underline (paramètre)

Fixer ce paramètre à TRUE activera le soulignement. Le soulignement est désactivé par défaut.

Overline (paramètre)

Fixer ce paramètre à TRUE activera surlignement. Le surlignement est désactivé par défaut.

Strikeout (paramètre)

Fixer ce paramètre à TRUE activera le texte barré. Le texte barré est désactivé par défaut.

8.106.73 **ps_string_geometry()** : Récupère les géométries d'une chaîne de caractères

array **ps_string_geometry** (resource psdoc , string text , int fontid , float size)

Cette fonction est similaire à [ps_stringwidth](#) mais retourne un tableau des dimensions.

8.106.74 **ps_stringwidth()** : Récupère la largeur d'une chaîne de caractères

float **ps_stringwidth** (resource psdoc , string text , int fontid , float size)

Calcule la largeur d'une chaîne de caractères en points si elle a été écrite dans la police et la taille donnée. Cette fonction requiert le fichier de police métrique "adobe" pour calculer la largeur précise. Si le crénage est activé, il sera pris en compte.

8.106.75 **ps_stroke()** : Dessine le chemin courant

bool **ps_stroke** (resource psdoc)

Dessine le chemin construit avec des fonctions de dessin comme [ps_lineto](#) précédemment appelées.

8.106.76 **ps_symbol_name()** : Récupère le nom d'un glyphe

string **ps_symbol_name** (resource psdoc , int ord , int fontid)

Cette fonction nécessite le fichier de police métrique "adobe" pour savoir quels glyphes sont disponibles.

8.106.77 **ps_symbol_width()** : Récupère la largeur d'un glyphe

float **ps_symbol_width** (resource psdoc , int ord , *int fontid* , *float size*)

Calcule la largeur d'un glyphe en points s'il a été écrit dans la police et la taille donnée. Cette fonction nécessite le fichier de police métrique "adobe" pour calculer la largeur précise.

8.106.78 **ps_symbol()** : Écrit un glyphe

bool **ps_symbol** (resource psdoc , int ord)

Écrit un glyphe à la position ord dans le vecteur d'encodage de la police de la police courante. L'encodage de la police pour une police peut être fixé lors du chargement de la police avec ps_findfont .

8.106.79 **ps_translate()** : Fixe une translation

bool **ps_translate** (resource psdoc , float x , float y)

Fixe un nouveau point initial d'un système coordonné.

8.107 Pspell

8.107.1 Introduction

La bibliothèque pspell vous permet de vérifier l'orthographe d'un mot, et suggérer des corrections.

8.107.2 Pré-requis

Pour compiler PHP avec le support pspell, vous aurez besoin de la bibliothèque aspell, disponible à <http://aspell.sourceforge.net/> .

8.107.3 Installation

Si vous avez installé les bibliothèques PSpell, vous devez simplement ajouter l'option `--with-pspell[=dir]` , lors de la compilation de PHP.

Note
Note pour les utilisateurs de Microsoft Windows
Le support pour Win32 est uniquement disponible depuis les versions PHP 4.3.3 et suivantes. Vous devez avoir Aspell 0.50 ou plus récent d'installé. Pour activer ce module sous les systèmes Windows, vous devez copier le fichier <code>aspell-15.dll</code> de votre dossier d'installation Aspell bin dans un dossier où PHP sera capable de le trouver. Les dossiers <code>C:\PHP</code> , <code>C:\WINNT\SYSTEM32</code> ou <code>C:\WINDOWS\SYSTEM32</code> sont de bons choix.

8.107.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.107.5 Types de ressources

Cette extension ne définit aucune ressource.

8.107.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

`PSPELL_FAST` ([entier](#))
`PSPELL_NORMAL` ([entier](#))
`PSPELL_BAD_SPELLERS` ([entier](#))
`PSPELL_RUN_TOGETHER` ([entier](#))

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)

- [Types de ressources](#)
- [Constantes pré-définies](#)
- [pspell_add_to_personal](#)
- [pspell_add_to_session](#)
- [pspell_check](#)
- [pspell_clear_session](#)
- [pspell_config_create](#)
- [pspell_config_data_dir](#)
- [pspell_config_dict_dir](#)
- [pspell_config_ignore](#)
- [pspell_config_mode](#)
- [pspell_config_personal](#)
- [pspell_config_repl](#)
- [pspell_config_runtogether](#)
- [pspell_config_save_repl](#)
- [pspell_new_config](#)
- [pspell_new_personal](#)
- [pspell_new](#)
- [pspell_save_wordlist](#)
- [pspell_store_replacement](#)
- [pspell_suggest](#)

8.107.8 [pspell_add_to_session\(\)](#) : Ajoute le mot au dictionnaire personnel de la session courante

bool [pspell_add_to_session](#) (int dictionary_link , string word)

[pspell_add_to_session](#) ajoute un mot au dictionnaire personnel associé à la version courante. C'est une fonction similaire à [pspell_add_to_personal](#) .

8.107.9 [pspell_check\(\)](#) : Vérifie un mot

bool [pspell_check](#) (int dictionary_link , string word)

[pspell_check](#) vérifie l'orthographe d'un mot et retourne TRUE si l'orthographe est correcte, FALSE sinon.

Exemple avec [pspell_check](#)

```
<?php
$pspell_link = pspell_new ("fr");

if (pspell_check($pspell_link, "testt")) {
    echo 'L\'orthographe est exacte';
} else {
    echo 'Désolé, mauvaise orthographe';
}
?>
```

8.107.10 [pspell_clear_session\(\)](#) : Remet à zéro la session courante

bool [pspell_clear_session](#) (int dictionary_link)

`pspell_clear_session` remet à zéro la session courante. Le dictionnaire personnel est vidé et, par exemple, si vous tentez de l'enregistrer avec `pspell_save_wordlist`, rien ne se passera.

Exemple avec `pspell_add_to_personal`

```
<?php
$pspell_config = pspell_config_create ("en");
pspell_config_personal ($pspell_config, "/var/dictionaries/custom.pws");
$pspell_link = pspell_new_config ($pspell_config);

pspell_add_to_personal ($pspell_link, "Vlad");
pspell_clear_session ($pspell_link);
pspell_save_wordlist ($pspell_link); // "Vlad" ne sera pas sauvé
?>
```

8.107.11 `pspell_config_create()` : Crée une configuration utilisée pour ouvrir un dictionnaire

int `pspell_config_create` (string language , string spelling , string jargon , string encoding)

`pspell_config_create` a une syntaxe similaire à celle de `pspell_new`. En fait, utiliser `pspell_config_create` suivi immédiatement par `pspell_new_config` produira exactement le même résultat. Cependant, après avoir créé une nouvelle configuration, vous pouvez aussi utiliser les fonctions `pspell_config_*` avant d'appeler `pspell_new_config` pour tirer profit des fonctionnalités avancées.

Le paramètre de langage `language` est le code de langue en deux lettres, défini dans la norme ISO 639, et deux lettres optionnelles ISO 3166, après un tiret ou un souligné (_).

Le paramètre d'orthographe `spelling` est nécessaire pour les langues qui ont plus d'une orthographe, comme l'anglais. Les valeurs reconnues sont alors 'american' (américain), 'british' (anglais), et 'canadian' (canadien).

Le paramètre de jargon `jargon` contient des informations supplémentaires pour distinguer deux dictionnaires distincts pour la même langue et le même paramètre d'orthographe `spelling`.

Le paramètre d'encodage `encoding` indique l'encodage attendu pour la réponse. Les valeurs valides sont : 'utf-8', 'iso8859-*', 'koi8-r', 'viscii', 'cp1252', 'machine unsigned 16', 'machine unsigned 32'. Ce paramètre n'a pas été testé de manière exhaustive, alors soyez prudent.

Le paramètre de mode est le mode de travail du vérificateur d'orthographe. Plusieurs modes sont disponibles :

- `PSPELL_FAST` - Mode rapide (moins de suggestions, plus de vitesse)
- `PSPELL_NORMAL` - Mode normal (plus de suggestions)
- `PSPELL_BAD_SPELLERS` - Mode lent (beaucoup plus de suggestions, moins de vitesse)

Pour plus d'informations et d'exemples, vérifiez le manuel `pspell` sur leur site web : <http://aspell.net/>

Exemple avec `pspell_config_create`

```
<?php
$pspell_config = pspell_config_create("en");
pspell_config_personal($pspell_config, "/var/dictionaries/custom.pws");
pspell_config_repl($pspell_config, "/var/dictionaries/custom.repl");
$pspell_link = pspell_new_personal($pspell_config, "en");
```

?>

8.107.12 `pspell_config_data_dir()` : Endroit où se trouvent les fichiers de données linguistiques

`bool pspell_config_data_dir (int conf , string directory)`

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.107.13 `pspell_config_dict_dir()` : Endroit où se trouve le fichier global des mots

`bool pspell_config_dict_dir (int conf , string directory)`

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.107.14 `pspell_config_ignore()` : Ignore les mots de moins de N caractères

`bool pspell_config_ignore (int dictionary_link , int n)`

`pspell_config_ignore` doit être utilisée avec une configuration avant d'appeler `pspell_new_config`. Cette fonction permet au vérificateur d'ignorer les mots trop courts. Les mots plus courts que n caractères seront ignorés.

Exemple avec `pspell_config_ignore`

```
<?php
$pspell_config = pspell_config_create ("en");
pspell_config_ignore($pspell_config, 5);
$pspell_link = pspell_new_config($pspell_config);
pspell_check($pspell_link, "abcd"); // Ce mot ne provoquera pas d'erreur
?>
```

8.107.15 `pspell_config_mode()` : Change le mode de suggestion

`bool pspell_config_mode (int dictionary_link , int mode)`

`pspell_config_mode` doit être appelée avant `pspell_new_config`. Cette fonction détermine le nombre de suggestions qui seront retournées par `pspell_suggest`.

Le paramètre de mode est le mode de travail du vérificateur d'orthographe. Plusieurs modes sont disponibles :

- `PSPELL_FAST` - Mode rapide (moins de suggestions, plus de vitesse)
- `PSPELL_NORMAL` - Mode normal (plus de suggestions)
- `PSPELL_BAD_SPELLERS` - Mode lent (beaucoup plus de suggestions, moins de vitesse)

Exemple avec `pspell_config_mode`

```
<?php
$pspell_config = pspell_config_create("en");
pspell_config_mode($pspell_config, PSPELL_FAST);
$pspell_link = pspell_new_config($pspell_config);
pspell_check($pspell_link, "thecat");
?>
```

8.107.16 `pspell_config_personal()` : Choisit le fichier qui contient le dictionnaire personnel

`bool pspell_config_personal (int dictionary_link , string file)`

`pspell_config_personal` doit être appelée dans une configuration avant d'appeler `pspell_new_config`. Le dictionnaire personnel sera chargé et utilisé en plus du dictionnaire standard, une fois que vous aurez appelé `pspell_new_config`. Si le fichier n'existe pas, il sera créé. Ce fichier sera aussi le fichier où `pspell_save_wordlist` sauvera le dictionnaire personnel ; il devra donc être accessible en écriture par PHP. Notez que cette fonction ne fonctionne pas avec les versions antérieures à `pspell .11.2` et `aspell .32.5`.

Exemple avec `pspell_config_personal`

```
<?php
$pspell_config = pspell_config_create("en");
pspell_config_personal($pspell_config, "/var/dictionaries/custom.pws");
$pspell_link = pspell_new_config($pspell_config);
pspell_check($pspell_link, "thecat");
?>
```

8.107.17 `pspell_config_repl()` : Choisit le fichier qui contient les paires de remplacement

`bool pspell_config_repl (int dictionary_link , string file)`

`pspell_config_repl` doit être appelée dans une configuration avant d'appeler `pspell_new_config`. Les paires de remplacement améliorent la qualité du vérificateur. Lorsqu'un mot est mal orthographié et qu'aucune suggestion valable n'est trouvée dans le dictionnaire, `pspell_store_replacement` sera utilisée pour enregistrer une paire de remplacement et `pspell_save_wordlist` pour sauver le dictionnaire avec les paires de remplacement. Ce fichier devra donc être accessible en écriture par PHP. Notez que cette fonction ne fonctionne pas avec les versions antérieures à `pspell .11.2` et `aspell .32.5`.

Exemple avec `pspell_config_repl`

```
<?php
$pspell_config = pspell_config_create("en");
pspell_config_personal($pspell_config, "/var/dictionaries/custom.pws");
pspell_config_repl($pspell_config, "/var/dictionaries/custom.repl");
$pspell_link = pspell_new_config($pspell_config);
pspell_check($pspell_link, "thecat");
?>
```

8.107.18 `pspell_config_runtogether()` : Considère deux mots accolés comme un composé

`bool pspell_config_runtogether (int dictionary_link , bool flag)`

8.107.15 `pspell_config_mode()` : Change le mode de suggestion

`pspell_config_runtogether` doit être appelée dans une configuration avant d'appeler `pspell_new_config`. Cette fonction indique si deux mots accolés doivent être traités comme un composé valide, même s'il devrait y avoir un espace entre ces deux mots. Modifier cette configuration n'affecte que les résultats retournés par `pspell_check`; `pspell_suggest` retournera toujours des suggestions.

Exemple avec `pspell_config_runtogether`

```
<?php
$pspell_config = pspell_config_create("en");
pspell_config_runtogether($pspell_config, true);
$pspell_link = pspell_new_config($pspell_config);
pspell_check($pspell_link, "thecat");
?>
```

8.107.19 `pspell_config_save_repl()` : Active la sauvegarde des paires de remplacement

`bool pspell_config_save_repl (int dictionary_link , bool flag)`

`pspell_config_save_repl` doit être appelée dans une configuration avant d'appeler `pspell_new_config`. Elle détermine si `pspell_save_wordlist` doit sauver les paires de remplacement avec le dictionnaire. Généralement, il n'y a pas besoin d'utiliser cette fonction car, si `pspell_config_repl` est utilisée, les paires de remplacement seront sauvées de toute façon, et, si ce n'est pas le cas, elles ne le seront pas. Ce fichier devra donc être accessible en écriture par PHP. Notez que cette fonction ne fonctionne pas avec les versions antérieures à `pspell .11.2` et `aspell .32.5`.

8.107.20 `pspell_new_config()` : Charge un nouveau dictionnaire

`int pspell_new_config (int config)`

`pspell_new_config` ouvre un nouveau dictionnaire et charge les paramètres spécifiés dans la configuration `config`, créée avec `pspell_config_create` et modifiée avec les fonctions `pspell_config_*`. Cette méthode vous donne le maximum de flexibilité, et dispose de toutes les fonctionnalités fournies par `pspell_new` et `pspell_new_personal`.

Le paramètre de configuration est celui qui a été retourné par `pspell_config_create` lors de la création de la configuration.

Exemple avec `pspell_new_config`

```
<?php
$pspell_config = pspell_config_create("en");
pspell_config_personal($pspell_config, "/var/dictionaries/custom.pws");
pspell_config_repl($pspell_config, "/var/dictionaries/custom.repl");
$pspell_link = pspell_new_personal($pspell_config);
?>
```

8.107.21 `pspell_new_personal()` : Charge un nouveau dictionnaire avec un dictionnaire personnel

`int pspell_new_personal (string personal , string language , string spelling , string jargon , string encoding , int mode)`

`pspell_new_personal` charge un nouveau dictionnaire avec un dictionnaire personnel, et retourne un identifiant de dictionnaire utilisé par d'autres fonctions `pspells`. Le dictionnaire peut être modifié et sauvé avec `pspell_save_wordlist`. Cependant, les paires de remplacement ne seront pas sauvées. Pour ce faire, vous devez créer une configuration qui utilise `pspell_config_create`, et choisir le fichier de destination du dictionnaire personnel avec `pspell_config_personal`, choisir le fichier de paire de remplacement avec `pspell_config_repl` et ouvrir un nouveau dictionnaire avec `pspell_new_config`.

Le paramètre `personal` spécifie le fichier où seront ajoutés les mots du dictionnaire personnel. Ce doit être un chemin absolu, qui commence par '/' car, sinon, il sera relatif à \$HOME, qui est "/root" sur la plupart des systèmes, et probablement pas ce que vous souhaitez.

Le paramètre de langage `language` est le code de langue en deux lettres, défini dans la norme ISO 639 et deux lettres optionnelles ISO 3166, après un tiret ou un souligné (_).

Le paramètre d'orthographe `spelling` est nécessaire pour les langues qui ont plus d'une orthographe, comme l'anglais. Les valeurs reconnues sont alors 'american' (américain), 'british' (anglais), et 'canadian' (canadien).

Le paramètre de jargon `jargon` contient des informations supplémentaires pour distinguer deux dictionnaires distincts pour la même langue et le même paramètre d'orthographe `spelling`.

Le paramètre d'encodage `encoding` indique l'encodage attendu pour la réponse. Les valeurs valides sont : 'utf-8', 'iso8859-*', 'koi8-r', 'viscii', 'cp1252', 'machine unsigned 16', 'machine unsigned 32'. Ce paramètre n'a pas été testé de manière exhaustive, alors soyez prudent.

Le paramètre de mode `mode` est le mode de travail du vérificateur d'orthographe. Plusieurs modes sont disponibles :

- `PSPELL_FAST` - Mode rapide (moins de suggestions, plus de vitesse)
- `PSPELL_NORMAL` - Mode normal (plus de suggestions)
- `PSPELL_BAD_SPELLERS` - Mode lent (beaucoup plus de suggestions, moins de vitesse)
- `PSPELL_RUN_TOGETHER` - Considère les mots attachés comme légaux. De ce fait, "lechat" sera un mot composé légal, bien qu'il devrait y avoir un espace entre les deux mots. Changer cette configuration n'affecte que le résultat retourné par `pspell_check`; `pspell_suggest` continuera de retourner les suggestions.

Le mode est un masque construit depuis les différentes constantes listées ci-dessous. Cependant, les constantes `PSPELL_FAST`, `PSPELL_NORMAL` et `PSPELL_BAD_SPELLERS` sont mutuellement exclusives, donc, vous ne devez sélectionner qu'une seule d'entre elles.

Pour plus d'informations et d'exemples, vérifiez le manuel `pspell` sur leur site web : <http://aspell.net/>

Exemple avec `pspell_new_personal`

```
<?php
$pspell_link = pspell_new_personal("/var/dictionaries/custom.pws",
                                "en", "", "", "",
                                PSPELL_FAST | PSPELL_RUN_TOGETHER);
?>
```

8.107.22 `pspell_new()` : Charge un nouveau dictionnaire

int `pspell_new` (string language , string spelling , string jargon , string encoding , int mode)

`pspell_new` ouvre un nouveau dictionnaire et retourne un identifiant de dictionnaire, pour être utilisé avec d'autres fonctions `pspell`.

Le paramètre de langue `spelling` est constitué des deux lettres du codage de langue ISO 639, et du codage optionnel de pays ISO 3166, séparés par un '_'.

Ce paramètre est nécessaire pour les langues qui ont plus d'une orthographe, comme l'anglais ou le français. Les valeurs reconnues sont ``americain'', ``britannique'', et ``canadien''.

Le paramètre `jargon` contient des informations supplémentaires pour distinguer deux listes de mots qui ont le même marquage de langue et d'orthographe.

Le paramètre `encoding` est le type d'encodage des mots. Les valeurs valides sont 'utf-8', 'iso8859-*', 'koi8-r', 'viscii', 'cp1252', 'machine unsigned 16', 'machine unsigned 32'.

Le paramètre `mode` est le mode de travail du vérificateur d'orthographe. Plusieurs modes sont disponibles :

- `PSPELL_FAST` - Mode rapide (moins de suggestions, plus de vitesse)
- `PSPELL_NORMAL` - Mode normal (plus de suggestions)
- `PSPELL_BAD_SPELLERS` - Mode lent (beaucoup plus de suggestions, moins de vitesse)
- `PSPELL_RUN_TOGETHER` - Considère que des mots accolés forment un composé autorisé. C'est-à-dire que "lechat" sera un composé valide. Cette option ne modifie que les résultats retournés par `pspell_check` ; `pspell_suggest` retournera toujours les mêmes suggestions.

`mode` est un champ de bit, construit à partir des constantes listées ci-dessus. Cependant, `PSPELL_FAST`, `PSPELL_NORMAL` et `PSPELL_BAD_SPELLERS` sont mutuellement exclusives : vous ne devez en utiliser qu'une seule en même temps.

Pour plus d'informations et d'exemples, reportez-vous au site <http://aspell.net/> (en anglais).

Exemple avec `pspell_new`

```
<?php
$pspell_link = pspell_new("en", "", "", "", PSPELL_FAST|PSPELL_RUN_TOGETHER);
?>
```

8.107.23 `pspell_save_wordlist()` : Sauve le dictionnaire personnel dans un fichier

bool `pspell_save_wordlist` (int dictionary_link)

`pspell_save_wordlist` sauve le dictionnaire personnel de la session courante. Le dictionnaire doit avoir été ouvert avec `pspell_new_personal`, et la localisation des fichiers doit avoir été spécifiée avec `pspell_config_personal` et (éventuellement) `pspell_config_repl`. Notez que cette fonction n'est pas disponible avec les versions antérieures à `pspell .11.2` et `aspell .32.5`.

Exemple `pspell_add_to_personal`

```
<?php
```

```

$pspell_config = pspell_config_create("en");
pspell_config_personal($pspell_config, "/tmp/dicts/newdict");
$pspell_link = pspell_new_config($pspell_config);

pspell_add_to_personal($pspell_link, "Vlad");
pspell_save_wordlist($pspell_link);
?>

```

8.107.24 pspell_store_replacement() : Enregistre une paire de remplacement pour un mot

bool **pspell_store_replacement** (int dictionary_link , string misspelled , string correct)

pspell_store_replacement enregistre une paire de remplacement pour un mot de façon à ce que cette suggestion soit retournée par pspell_suggest plus tard. Pour pouvoir utiliser cette fonction, vous devez utiliser pspell_new_personal pour ouvrir le dictionnaire. Pour pouvoir sauver tout le temps les paires de remplacement, vous devez utiliser pspell_config_personal et pspell_config_repl pour indiquer le lieu de sauvegarde des dictionnaires personnels, et pspell_save_wordlist pour enregistrer les modifications sur le disque. Ce fichier devra donc être accessible en écriture par PHP. Notez que cette fonction ne fonctionne pas avec les versions antérieures à pspell .11.2 et aspell .32.5.

Exemple avec pspell_store_replacement

```

<?php
$pspell_config = pspell_config_create("en");
pspell_config_personal($pspell_config, "/var/dictionaries/custom.pws");
pspell_config_repl($pspell_config, "/var/dictionaries/custom.repl");
$pspell_link = pspell_new_config($pspell_config);

pspell_store_replacement($pspell_link, $misspelled, $correct);
pspell_save_wordlist($pspell_link);
?>

```

8.107.25 pspell_suggest() : Suggère une orthographe

array **pspell_suggest** (int dictionary_link , string word)

pspell_suggest retourne un tableau de suggestions pour le mot word .

Exemple avec pspell_suggest

```

<?php
$pspell_link = pspell_new("en");

if (!pspell_check($pspell_link, "testt")) {
    $suggestions = pspell_suggest($pspell_link, "testt");

    foreach ($suggestions as $suggestion) {
        echo "Orthographes suggérées : $suggestion<br />";
    }
}
?>

```

8.108 qtdom

Attention
Ce module est EXPERIMENTAL . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez ce module à vos risques et périls.
Note
Cette extension n'est pas disponible sur les plates-formes Windows.
Note
Cette extension a été déplacée dans le module <u>PECL</u> et ne sera plus intégrée dans PHP à partir de PHP 5.0.0.

8.108.1 Pré-requis

Vous aurez besoin de la bibliothèque Qt-library >=2.2.0.

8.108.2 Installation

Pour ajouter le support de qtdom dans PHP, ajoutez l'option --with-qtdom lors de la compilation.

- [Pré-requis](#)
- [Installation](#)
- [qdom_error](#)
- [qdom_tree](#)

8.108.4 qdom_tree() : Crée une arbre à partir d'une chaîne XML

QDomDocument **qdom_tree** (string doc)

Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.109 Radius

8.109.1 Introduction

Ce paquetage est basé sur la libradius de FreeBSD. Ce PECL ajoute un support complet pour l'identification Radius ([RFC 2865](#)) et les comptes Radius ([RFC 2866](#)). Ce paquetage est disponible pour Unix (testé sous FreeBSD et Linux) et pour Windows.

Note
Une description exacte de la libradius peut être trouvée ici . Une description détaillée du fichier de configuration peut être trouvée ici .

8.109.2 Installation

Comment installer le paquetage ?

- Décompresser le paquetage (habituellement dans php4/ext)
- Renommer radius-x.x en radius
- Exécuter ./buildconf dans php4
- Exécuter ./configure --enable-radius
- make; make install

ou, si vous voulez l'avoir en .so :

- Décompresser le paquetage
- Exécuter phpize dans le dossier radius-x.x
- Exécuter ./configure dans le dossier radius-x.x
- make; make install

Pour Windows, je vous recommande d'utiliser la bibliothèque php_radius.dll depuis <http://snaps.php.net/> . Vous pouvez télécharger cette extension PECL séparée sur les différentes pages de publication PECL (sélectionnez le bon entrepôt en fonction de votre version de PHP) : [PECL pour PHP 4.3.x](#) , [PECL pour PHP 5.0.x](#) or [PECL instable](#) .

8.109.3 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

```

RADIUS_ACCESS_REQUEST ( )
    Demande d'identification
RADIUS_ACCESS_ACCEPT ( )
    Accès accepté
RADIUS_ACCESS_REJECT ( )
    Accès rejeté
RADIUS_ACCOUNTING_REQUEST ( )
    Demande de compte
RADIUS_ACCOUNTING_RESPONSE ( )
    Réponse du compte
RADIUS_ACCESS_CHALLENGE ( )
    Défi d'accès

```

RADIUS_USER_NAME (string)
 Nom d'utilisateur

RADIUS_USER_PASSWORD (string)
 Mot de passe

RADIUS_CHAP_PASSWORD (string)
 Chap Password: chappass = md5(ident + plaintextpass + challenge)

RADIUS_NAS_IP_ADDRESS (string)
 Adresse IP NAS

RADIUS_NAS_PORT (int)
 Port NAS

RADIUS_SERVICE_TYPE (int)
 Type de service :

- RADIUS_LOGIN
- RADIUS_FRAMED
- RADIUS_CALLBACK_LOGIN
- RADIUS_CALLBACK_FRAMED
- RADIUS_OUTBOUND
- RADIUS_ADMINISTRATIVE
- RADIUS_NAS_PROMPT
- RADIUS_AUTHENTICATE_ONLY
- RADIUS_CALLBACK_NAS_PROMPT

RADIUS_FRAMED_PROTOCOL (int)
 Protocole encadré :

- RADIUS_PPP
- RADIUS_SLIP
- RADIUS_ARAP
- RADIUS_GANDALF
- RADIUS_XYLOGICS

RADIUS_FRAMED_IP_ADDRESS (string)
 Adresse IP

RADIUS_FRAMED_IP_NETMASK (string)
 Netmasque

RADIUS_FRAMED_ROUTING (int)
 Routing

RADIUS_FILTER_ID (string)
 Filter ID

RADIUS_FRAMED_MTU (int)
 MTU

RADIUS_FRAMED_COMPRESSION (int)
 Compression :

- RADIUS_COMP_NONE
- RADIUS_COMP_VJ
- RADIUS_COMP_IPXHDR

RADIUS_LOGIN_IP_HOST (string)
 Login IP Host

RADIUS_LOGIN_SERVICE (int)
 Login Service

RADIUS_LOGIN_TCP_PORT (int)
 Login TCP Port

RADIUS_REPLY_MESSAGE (string)
 Message de réponse

RADIUS_CALLBACK_NUMBER (string)
 Numéro de Callback

RADIUS_CALLBACK_ID (string)

ID de Callback
 RADIUS_FRAMED_ROUTE (string)
 Framed Route
 RADIUS_FRAMED_IPX_NETWORK (string)
 Framed IPX Network
 RADIUS_STATE (string)
 Statut
 RADIUS_CLASS (int)
 Classe
 RADIUS_VENDOR_SPECIFIC (int)
 Attribut spécifique au vendeur
 RADIUS_SESSION_TIMEOUT (int)
 Timeout de la session
 RADIUS_IDLE_TIMEOUT (int)
 Idle timeout
 RADIUS_TERMINATION_ACTION (int)
 Termination action
 RADIUS_CALLED_STATION_ID (int)
 Called Station Id
 RADIUS_CALLING_STATION_ID (string)
 Calling Station Id
 RADIUS_NAS_IDENTIFIER (int)
 NAS ID
 RADIUS_PROXY_STATE (int)
 Proxy State
 RADIUS_LOGIN_LAT_SERVICE (int)
 Login LAT Service
 RADIUS_LOGIN_LAT_NODE (int)
 Login LAT Node
 RADIUS_LOGIN_LAT_GROUP (int)
 Login LAT Group
 RADIUS_FRAMED_APPLETALK_LINK (int)
 Framed Appletalk Link
 RADIUS_FRAMED_APPLETALK_NETWORK (int)
 Framed Appletalk Network
 RADIUS_FRAMED_APPLETALK_ZONE (int)
 Framed Appletalk Zone
 RADIUS_CHAP_CHALLENGE (string)
 Challenge
 RADIUS_NAS_PORT_TYPE (int)
 Type du port NAS :
 RADIUS_ASYNC
 RADIUS_SYNC
 RADIUS_ISDN_SYNC
 RADIUS_ISDN_ASYNC_V120
 RADIUS_ISDN_ASYNC_V110
 RADIUS_VIRTUAL
 RADIUS_PIAFS
 RADIUS_HDLC_CLEAR_CHANNEL
 RADIUS_X_25
 RADIUS_X_75
 RADIUS_G_3_FAX
 RADIUS_SDSL
 RADIUS_ADSL_CAP

```

RADIUS_ADSL_DMT
RADIUS_IDSL
RADIUS_ETHERNET
RADIUS_XDSL
RADIUS_CABLE
RADIUS_WIRELESS_OTHER
RADIUS_WIRELESS_IEEE_802_11
RADIUS_PORT_LIMIT ( int )
    Port Limit
RADIUS_LOGIN_LAT_PORT ( int )
    Login LAT Port
RADIUS_CONNECT_INFO ( string )
    Connect info
RADIUS_ACCT_STATUS_TYPE ( int )
    Type de statut du compte :
        RADIUS_START
        RADIUS_STOP
        RADIUS_ACCOUNTING_ON
        RADIUS_ACCOUNTING_OFF
RADIUS_ACCT_DELAY_TIME ( int )
    Accounting delay time
RADIUS_ACCT_INPUT_OCTETS ( int )
    Accounting input bytes
RADIUS_ACCT_OUTPUT_OCTETS ( int )
    Accounting output bytes
RADIUS_ACCT_SESSION_ID ( int )
    Accounting session ID
RADIUS_ACCT_AUTHENTIC ( int )
    Accounting authentic, one of:
        RADIUS_AUTH_RADIUS
        RADIUS_AUTH_LOCAL
        RADIUS_AUTH_REMOTE
RADIUS_ACCT_SESSION_TIME ( int )
    Accounting session time
RADIUS_ACCT_INPUT_PACKETS ( int )
    Accounting input packets
RADIUS_ACCT_OUTPUT_PACKETS ( int )
    Accounting output packets
RADIUS_ACCT_TERMINATE_CAUSE ( int )
    Accounting terminate cause, one of:
        RADIUS_TERM_USER_REQUEST
        RADIUS_TERM_LOST_CARRIER
        RADIUS_TERM_LOST_SERVICE
        RADIUS_TERM_IDLE_TIMEOUT
        RADIUS_TERM_SESSION_TIMEOUT
        RADIUS_TERM_ADMIN_RESET
        RADIUS_TERM_ADMIN_REBOOT
        RADIUS_TERM_PORT_ERROR
        RADIUS_TERM_NAS_ERROR
        RADIUS_TERM_NAS_REQUEST
        RADIUS_TERM_NAS_REBOOT
        RADIUS_TERM_PORT_UNNEEDED
        RADIUS_TERM_PORT_PREEMPTED
        RADIUS_TERM_PORT_SUSPENDED

```

RADIUS_TERM_SERVICE_UNAVAILABLE
RADIUS_TERM_CALLBACK
RADIUS_TERM_USER_ERROR
RADIUS_TERM_HOST_REQUEST
RADIUS_ACCT_MULTI_SESSION_ID (string)
Accounting multi session ID
RADIUS_ACCT_LINK_COUNT (int)
Accounting link count
RADIUS_VENDOR_MICROSOFT (int)
Microsoft specific vendor attributes ([RFC 2548](#)), one of:
RADIUS_MICROSOFT_MS_CHAP_RESPONSE
RADIUS_MICROSOFT_MS_CHAP_ERROR
RADIUS_MICROSOFT_MS_CHAP_PW_1
RADIUS_MICROSOFT_MS_CHAP_PW_2
RADIUS_MICROSOFT_MS_CHAP_LM_ENC_PW
RADIUS_MICROSOFT_MS_CHAP_NT_ENC_PW
RADIUS_MICROSOFT_MS_MPPE_ENCRYPTION_POLICY
RADIUS_MICROSOFT_MS_MPPE_ENCRYPTION_TYPES
RADIUS_MICROSOFT_MS_RAS_VENDOR
RADIUS_MICROSOFT_MS_CHAP_DOMAIN
RADIUS_MICROSOFT_MS_CHAP_CHALLENGE
RADIUS_MICROSOFT_MS_CHAP_MPPE_KEYS
RADIUS_MICROSOFT_MS_BAP_USAGE
RADIUS_MICROSOFT_MS_LINK_UTILIZATION_THRESHOLD
RADIUS_MICROSOFT_MS_LINK_DROP_TIME_LIMIT
RADIUS_MICROSOFT_MS_MPPE_SEND_KEY
RADIUS_MICROSOFT_MS_MPPE_RECV_KEY
RADIUS_MICROSOFT_MS_RAS_VERSION
RADIUS_MICROSOFT_MS_OLD_ARAP_PASSWORD
RADIUS_MICROSOFT_MS_NEW_ARAP_PASSWORD
RADIUS_MICROSOFT_MS_ARAP_PASSWORD_CHANGE_REASON
RADIUS_MICROSOFT_MS_FILTER
RADIUS_MICROSOFT_MS_ACCT_AUTH_TYPE
RADIUS_MICROSOFT_MS_ACCT_EAP_TYPE
RADIUS_MICROSOFT_MS_CHAP2_RESPONSE
RADIUS_MICROSOFT_MS_CHAP2_SUCCESS
RADIUS_MICROSOFT_MS_CHAP2_PW
RADIUS_MICROSOFT_MS_PRIMARY_DNS_SERVER
RADIUS_MICROSOFT_MS_SECONDARY_DNS_SERVER
RADIUS_MICROSOFT_MS_PRIMARY_NBNS_SERVER
RADIUS_MICROSOFT_MS_SECONDARY_NBNS_SERVER
RADIUS_MICROSOFT_MS_ARAP_CHALLENGE

8.109.4 Démarrage rapide

Comment commencer ?

- Récupérer une ressource Radius
- Configurer la bibliothèque
- Créer une demande
- Ajouter des attributs
- Envoyer la demande
- Recevoir les attributs

- Fermer la ressource Radius (optionnel)

Regardez également les exemples fournis avec ce paquetage.

Ce paquetage contient un script PHP d'exemple. Ce script montre comment s'identifier avec Radius en utilisant PAP ou CHAP (md5). Si vous voulez vous identifier avec un serveur Microsoft Radius, alors il n'est pas possible d'utiliser CHAP (md5). Si vous voulez vous identifier avec un serveur Microsoft Radius, vous devez utiliser MS-CHAPv1 ou MS-CHAPv2 mais c'est plus compliqué, parce que vous avez besoin de md4, sha1 et DES pour générer de bonnes données. L'exemple fourni montre toutes les méthodes d'identification, incluant MS-CHAPv1 et MS-CHAPv2. Pour faire fonctionner MS-CHAP, vous aurez besoin des extensions [mcrypt](#) et [mhash](#), et depuis la version 1.2 de ce paquetage, l'extension mcrypt n'est plus nécessaire.

8.109.5 Contact

Si vous avez des commentaires, des corrections de bogues, des améliorations ou si vous voulez aider dans le développement, vous pouvez envoyer un email à mbretter@php.net. Les binaires Windows peuvent être téléchargés [ici](#).

- [Introduction](#)
- [Installation](#)
- [Constantes pré-définies](#)
- [Démarrage rapide](#)
- [Contact](#)
- [radius_acct_open](#)
- [radius_add_server](#)
- [radius_auth_open](#)
- [radius_close](#)
- [radius_config](#)
- [radius_create_request](#)
- [radius_cvt_addr](#)
- [radius_cvt_int](#)
- [radius_cvt_string](#)
- [radius_demangle_mppe_key](#)
- [radius_demangle](#)
- [radius_get_attr](#)
- [radius_get_vendor_attr](#)
- [radius_put_addr](#)
- [radius_put_attr](#)
- [radius_put_int](#)
- [radius_put_string](#)
- [radius_put_vendor_addr](#)
- [radius_put_vendor_attr](#)
- [radius_put_vendor_int](#)
- [radius_put_vendor_string](#)
- [radius_request_authenticator](#)
- [radius_send_request](#)
- [radius_server_secret](#)
- [radius_strerror](#)

8.109.7 `radius_add_server()` : Ajoute un serveur

bool `radius_add_server` (resource `radius_handle` , string `hostname` , int `port` , string `secret` , int `timeout` , int `max_tries`)

`radius_add_server` peut être utilisé plusieurs fois, et il peut être utilisé avec la fonction `radius_config` . Tout au plus, 10 serveurs peuvent être spécifiés. Lorsque plusieurs serveurs sont fournis, ils sont essayés à la façon round-robin tant qu'une réponse valide n'est pas reçue, ou tant que la limite `max_tries` de chaque serveur n'est pas atteinte.

8.109.8 `radius_auth_open()` : Crée une ressource Radius pour l'identification

resource `radius_auth_open` ()

8.109.9 `radius_close()` : Libère toutes les ressources

bool `radius_close` (resource `radius_handle`)

Il n'est pas nécessaire d'appeler cette fonction car PHP libère toutes les ressources à la fin de chaque demande.

8.109.10 `radius_config()` : Demande à la bibliothèque de lire un fichier de configuration donné

bool `radius_config` (resource `radius_handle` , string `file`)

Avant toute demande Radius, la bibliothèque doit connaître le serveur à contacter. La manière simple de configurer la bibliothèque est d'appeler la fonction `radius_config` . `radius_config` demande à la bibliothèque de lire un fichier de configuration dont le format est décrit sur la page `radius.conf` .

8.109.11 `radius_create_request()` : Crée une demande de compte ou d'identification

bool `radius_create_request` (resource `radius_handle` , int `type`)

Une demande Radius consiste en du code spécifiant une demande spécifique, ainsi que zéro ou plusieurs attributs qui fournissent des informations supplémentaires. Pour commencer à construire une nouvelle demande, appelez la fonction `radius_create_request` .

Note

Attention : Vous devez appeler cette fonction avant de passer n'importe quel argument !

8.109.12 `radius_cvt_addr()` : Convertit des données brutes en adresse IP

string `radius_cvt_addr` (string `data`)

8.109.13 `radius_cvt_int()` : Convertit des données brutes en entier

int `radius_cvt_int` (string data)

8.109.14 `radius_cvt_string()` : Convertit des données brutes en chaîne de caractères

string `radius_cvt_string` (string data)

8.109.15 `radius_demangle_mppe_key()` : Dérive les clés mppe depuis des données

string `radius_demangle_mppe_key` (resource radius_handle , string mangled)

Lors de l'utilisation de MPPE avec MS-CHAPv2, les clés reçues et envoyées sont "séchées" (voir la [RFC 2548](#)), cependant, cette fonction est utile, car je ne sais pas s'il y a ou pas une implémentation de PPTP-MPPE dans PHP.

8.109.16 `radius_demangle()` : Assèche des données

string `radius_demangle` (resource radius_handle , string mangled)

Quelques données (mots de passe, clés MS-CHAPv1 MPPE) sont "séchées" pour des raisons de sécurité et doivent être "asséchées" avant de pouvoir les utiliser.

8.109.17 `radius_get_attr()` : Extrait un attribut

mixed `radius_get_attr` (resource radius_handle)

Comme les demandes Radius, chaque réponse doit contenir zéro ou plusieurs attributs. Après la réception d'une réponse avec succès par la fonction `radius_send_request`, ces attributs peuvent être extraits un par un en utilisant la fonction `radius_get_attr`. A chaque appel de `radius_get_attr`, le prochain attribut est récupéré depuis la réponse courante.

8.109.18 `radius_get_vendor_attr()` : Extrait un attribut spécifique au vendeur

array `radius_get_vendor_attr` (string data)

Si `radius_get_attr` retourne `RADIUS_VENDOR_SPECIFIC`, `radius_get_vendor_attr` peut être appelé pour déterminer le vendeur.

8.109.19 `radius_put_addr()` : Attache une adresse IP en tant qu'attribut

bool `radius_put_addr` (resource radius_handle , int type , string addr)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.109.20 radius_put_attr() : Attache un attribut binaire

bool radius_put_attr (resource radius_handle , int type , string value)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.109.21 radius_put_int() : Attache un attribut entier

bool radius_put_int (resource radius_handle , int type , int value)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.109.22 radius_put_string() : Attache un attribut chaîne de caractères

bool radius_put_string (resource radius_handle , int type , string value)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.109.23 radius_put_vendor_addr() : Attache un attribut IP-Address spécifique à un vendeur

bool radius_put_vendor_addr (resource radius_handle , int vendor , int type , string addr)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.109.24 radius_put_vendor_attr() : Attache un attribut binaire à un vendeur spécifique

bool radius_put_vendor_attr (resource radius_handle , int vendor , int type , string value)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.109.25 radius_put_vendor_int() : Attache un attribut entier à un vendeur spécifique

bool radius_put_vendor_int (resource radius_handle , int vendor , int type , int value)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.109.26 `radius_put_vendor_string()` : Attache un attribut sous la forme d'une chaîne à un vendeur spécifique

bool `radius_put_vendor_string` (resource `radius_handle` , int `vendor` , int `type` , string `value`)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.109.27 `radius_request_authenticator()` : Retourne l'identificateur demandé

string `radius_request_authenticator` (resource `radius_handle`)

L'identificateur demandé est nécessaire pour le rétablissement des données comme les mots de passe et les clés de cryptages.

8.109.28 `radius_send_request()` : Envoie la demande et attende une réponse

int `radius_send_request` (resource `radius_handle`)

Après qu'une demande Radius ait été construite, elle est envoyée grâce à la fonction `radius_send_request` .

La fonction `radius_send_request` envoie la requête et attend une réponse valide, re-essayant à la façon round-robin autant que nécessaire.

8.109.29 `radius_server_secret()` : Retourne le secret partagé

string `radius_server_secret` (resource `radius_handle`)

Le secret partagé est nécessaire pour le rétablissement des données comme les mots de passe et les clés de cryptages.

8.109.30 `radius_strerror()` : Retourne un message d'erreur

string `radius_strerror` (resource `radius_handle`)

Si une fonction Radius échoue, alors, elle enregistre un message d'erreur. Ce message d'erreur peut être récupéré avec cette fonction.

8.110 Rar

8.110.1 Introduction

Rar est un mécanisme d'archivage puissant et efficace, créé par Eugene Roshal. Cette extension vous donne la possibilité de lire des archives Rar, mais pas de créer des archives Rar, car ce n'est pas supporté par la bibliothèque, et c'est explicitement interdit par la licence.

Plus d'informations sur Rar et UnRar sont disponibles sur <http://www.rarlabs.com/> .

8.110.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.110.3 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.110.4 Installation

Rar est actuellement disponible via PECL <http://pecl.php.net/package/rar> .

Vous pouvez utiliser l'installeur PEAR pour installer l'extension Rar, avec la commande suivante :
pear -v install rar .

Vous pouvez aussi télécharger le paquet tar.gz, et installer Rar manuellement :

Installation de Rar
<pre>gunzip rar-xxx.tgz tar -xvf rar-xxx.tar cd rar-xxx phpize ./configure && make && make install</pre>

Les utilisateurs Windows peuvent télécharger l'extension DLL php_rar.dll ici :
http://snaps.php.net/win32/PECL_STABLE/ .

8.110.5 Types de ressources

Il y a une ressource, utilisée par l'extension Rar : un pointeur de fichier, retourné par [rar_open](#) .

8.110.6 Constantes pré-définies

- RAR_HOST_MSDOS ([entier](#))
- RAR_HOST_OS2 ([entier](#))
- RAR_HOST_WIN32 ([entier](#))
- RAR_HOST_UNIX ([entier](#))
- RAR_HOST_BEOS ([entier](#))

8.110.7 Exemples

Exemple avec l'extension Rar

```
<?php

$rar_file = rar_open('exemple.rar')
    or die("Impossible d'ouvrir l'archive Rar");

$entries = rar_list($rar_file);

foreach ($entries as $entry) {
    echo 'Nom du fichier : ' . $entry->getName() . "\n";
    echo 'Taille du fichier : ' . $entry->getPackedSize() . "\n";
    echo 'Taille du fichier décompressé : ' . $entry->getUnpackedSize() . "\n";

    $entry->extract('/dossier/extraction/');
}

rar_close($rar_file);

?>
```

Cet exemple ouvre une archive Rar, et extrait chaque élément du dossier.

- [Introduction](#)
- [Pré-requis](#)
- [Configuration à l'exécution](#)
- [Installation](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [rar_close](#)
- [rar_entry_get](#)
- [Rar::extract](#)
- [Rar::getAttr](#)
- [Rar::getCrc](#)
- [Rar::getFileTime](#)
- [Rar::getHostOs](#)
- [Rar::getMethod](#)
- [Rar::getName](#)
- [Rar::getPackedSize](#)
- [Rar::getUnpackedSize](#)
- [Rar::getVersion](#)
- [rar_list](#)
- [rar_open](#)

8.110.9 rar_entry_get() : Lit un élément dans l'archive Rar

RarEntry **rar_entry_get** (resource rar_file , string entry_name)

rar_entry_get lit l'objet décrivant l'élément d'archive Rar courant.

Exemple avec [rar_entry_get](#)

```
<?php

$rar_file = rar_open('example.rar')
```

```

    or die("Impossible d'ouvrir l'archive Rar");
$entry = rar_entry_get($rar_file, 'Dir/file.txt')
    or die("Impossible de trouver cet élément");

print_r($entry);

?>

```

rar_get_entry retourne un objet ou bien FALSE en cas d'erreur.

8.110.10 Rar::extract() : Extrait un élément d'une archive Rar

bool **Rar::extract** (string dir , string filepath)

Rar::extract extrait les données de l'élément courant dans le dossier dir . Un nouveau fichier est créé dans le dossier dir , avec le nom de l'élément d'archive. Si filepath est spécifié à la place de dir , Rar::extract extraira les données dans le fichier spécifié.

Exemple avec Rar::extract

```

<?php

$rar_file = rar_open('example.rar')
    or die("Impossible d'ouvrir l'archive Rar");

$entry = rar_entry_get($rar_file, 'Dir/file.txt')
    or die("Impossible de trouver cet élément");

$entry->extract('/dir/to'); // crée le fichier /dir/to/Dir/file.txt
$entry->extract(false, '/dir/to/new_name.txt'); // crée le fichier /dir/to/new_name.txt

?>

```

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.110.11 Rar::getAttr() : Lit les attributs d'un élément d'archive Rar

int **Rar::getAttr** (void)

Rar::getAttr retourne les attributs de l'élément courant de l'archive Rar.

Exemple avec Rar::getAttr

```

<?php

$rar_file = rar_open('example.rar')
    or die("Impossible d'ouvrir l'archive Rar");

$entry = rar_entry_get($rar_file, 'Dir/file.txt')
    or die("Impossible de trouver cet élément");

$host_os = $entry->getHostOs();
$attr = $entry->getAttr();

switch($host_os) {
    case RAR_HOST_MSDOS:
    case RAR_HOST_OS2:
    case RAR_HOST_WIN32:
    case RAR_HOST_MACOS:

```

```

        printf("%c%c%c%c%c%c\n",
            ($attr & 0x08) ? 'V' : '.',
            ($attr & 0x10) ? 'D' : '.',
            ($attr & 0x01) ? 'R' : '.',
            ($attr & 0x02) ? 'H' : '.',
            ($attr & 0x04) ? 'S' : '.',
            ($attr & 0x20) ? 'A' : '.');
        break;
    case RAR_HOST_UNIX:
    case RAR_HOST_BEOS:
        switch ($attr & 0xF000)
        {
            case 0x4000:
                printf("d");
                break;
            case 0xA000:
                printf("l");
                break;
            default:
                printf("-");
                break;
        }
        printf("%c%c%c%c%c%c%c%c%c\n",
            ($attr & 0x0100) ? 'r' : '-',
            ($attr & 0x0080) ? 'w' : '-',
            ($attr & 0x0040) ? (($attr & 0x0800) ? 's':'x') : (($attr & 0x0800) ? 'S':'-'),
            ($attr & 0x0020) ? 'r' : '-',
            ($attr & 0x0010) ? 'w' : '-',
            ($attr & 0x0008) ? (($attr & 0x0400) ? 's':'x') : (($attr & 0x0400) ? 'S':'-'),
            ($attr & 0x0004) ? 'r' : '-',
            ($attr & 0x0002) ? 'w' : '-',
            ($attr & 0x0001) ? 'x' : '-');
        break;
    }
}
rar_close($rar_file);
?>

```

Rar::getAttr retourne FALSE en cas d'erreur.

Voir aussi Rar::getHostOs .

8.110.12 Rar::getCrc() : Lit la somme de contrôle d'un élément d'archive Rar

int **Rar::getCrc** (void)

Rar::getCrc lit la somme de contrôle de l'élément d'archive Rar courant.

Exemple avec Rar::getCrc

```

<?php
?>

```

Rar::getCrc retourne FALSE en cas d'erreur.

8.110.13 Rar::getFileTime() : Lit la date de dernière modification d'un élément d'archive Rar

string **Rar::getFileTime** (void)

Rar::getFileTime retourne la date de modification de l'élément courant, au format YYYY-MM-DD HH:II:SS .

Exemple avec Rar::getFileTime

```
<?php
$rar_file = rar_open('example.rar')
    or die("Impossible d'ouvrir l'archive Rar");

$entry = rar_entry_get($rar_file, 'Dir/file.txt')
    or die("Impossible de trouver cet élément");

echo "Modification de " . $entry->getName() . " : " . $entry->getFileTime();

?>
```

Rar::getFileTime retourne FALSE en cas d'erreur.

8.110.14 Rar::getHostOs() : Lit l'OS d'un élément d'archive Rar

int **Rar::getHostOs** (void)

Rar::getHostOs lit l'OS de l'élément d'archive Rar courant.

Exemple avec Rar::getHostOs

```
<?php
$rar_file = rar_open('example.rar')
    or die("Impossible d'ouvrir l'archive Rar");

$entry = rar_entry_get($rar_file, 'Dir/file.txt')
    or die("Impossible de trouver cet élément");

switch ($entry->getHostOs()) {
    case RAR_HOST_MSDOS:
        echo "MS-DOS\n";
        break;
    case RAR_HOST_OS2:
        echo "OS2\n";
        break;
    case RAR_HOST_WIN32:
        echo "Win32\n";
        break;
    case RAR_HOST_MACOS:
        echo "MacOS\n";
        break;
    case RAR_HOST_UNIX:
        echo "Unix/Linux\n";
        break;
    case RAR_HOST_BEOS:
        echo "BeOS\n";
        break;
}

?>
```

Rar::getHostOs retourne FALSE en cas d'erreur.

8.110.15 Rar::getMethod() : Lit la méthode de compression d'un élément d'archive Rar

int **Rar::getMethod** (void)

Rar::getMethod retourne un nombre représentant la méthode utilisée lors de la compression Rar de l'élément d'archive courant.

Exemple avec Rar::getMethod

```
<?php
$rar_file = rar_open('example.rar')
    or die("Impossible d'ouvrir l'archive Rar");

$entry = rar_entry_get($rar_file, 'Dir/file.txt')
    or die("Impossible de trouver cet élément");

echo "Numéro de la méthode : " . $entry->getMethod();

?>
```

Rar::getMethod retourne FALSE en cas d'erreur.

8.110.16 Rar::getName() : Lit le nom de l'élément d'archive Rar

string **Rar::getName** (void)

Rar::getName retourne le nom complet de l'élément d'archive.

Exemple avec Rar::getName

```
<?php
$rar_file = rar_open('example.rar')
    or die("Impossible d'ouvrir l'archive Rar");

$entry = rar_entry_get($rar_file, 'Dir/file.txt')
    or die("Impossible de trouver cet élément");

echo "Nom de l'élément : " . $entry->getName();

?>
```

Rar::getName returns FALSE on error.

8.110.17 Rar::getPackedSize() : Lit la taille compressée d'un élément d'archive Rar

int **Rar::getPackedSize** (void)

Rar::getPackedSize lit la taille compressée de l'élément d'archive Rar.

Exemple avec Rar::getPackedSize

```
<?php
$rar_file = rar_open('example.rar')
    or die("Impossible d'ouvrir l'archive Rar");

$entry = rar_entry_get($rar_file, 'Dir/file.txt')
    or die("Impossible de trouver cet élément");

echo "Taille compressée de " . $entry->getName() . " = " . $entry->getPackedSize() . " octets";
?>
```

Rar::getPackedSize returns FALSE on error.

8.110.18 Rar::getUnpackedSize() : Lit la taille décompressée d'un élément d'archive Rar

int **Rar::getUnpackedSize** (void)

Rar::getUnpackedSize lit la taille décompressée de l'élément d'archive Rar.

Exemple avec Rar::getUnpackedSize

```
<?php
$rar_file = rar_open('example.rar')
    or die("Impossible d'ouvrir l'archive Rar");

$entry = rar_entry_get($rar_file, 'Dir/file.txt')
    or die("Impossible de trouver cet élément");

echo "Taille décompressée de " . $entry->getName() . " = " . $entry->getUnpackedSize() . " octets";
?>
```

Rar::getUnpackedSize returns FALSE on error.

8.110.19 Rar::getVersion() : Lit la version utilisée pour créer le dernier élément Rar

int **Rar::getVersion** (void)

Rar::getVersion lit la version utilisée pour créer le dernier élément Rar.

Exemple avec Rar::getVersion

```
<?php
$rar_file = rar_open('example.rar')
    or die("Impossible d'ouvrir l'archive Rar");

$entry = rar_entry_get($rar_file, 'Dir/file.txt')
    or die("Impossible de trouver cet élément");

echo "Version de Rar (WinRAR) utilisée : " . $entry->getVersion();
?>
```

Rar::getVersion retourne FALSE en cas d'erreur.

8.110.20 rar_list() : Liste les éléments de l'archive Rar

array **rar_list** (resource rar_file)

rar_list liste les éléments de l'archive Rar rar_file .

Exemple avec rar_list

```
<?php
$rar_file = rar_open('example.rar')
    or die("Impossible d'ouvrir l'archive Rar");
$entries_list = rar_list($rar_file);
print_r($entries_list);
?>
```

rar_list retourne un tableau d'éléments, ou FALSE en cas d'erreur.

8.110.21 rar_open() : Ouvre une archive Rar

resource **rar_open** (string filename , string password)

rar_open ouvre l'archive Rar filename et retourne une ressource Rar.

rar_open retourne une ressource Rar ou bien FALSE en cas d'erreur.

8.111 Readline (GNU)

8.111.1 Introduction

Les fonctions readline implémentent une interface avec la bibliothèque GNU Readline. Ces fonctions fournissent une ligne de commande éditable, un peu comme lorsque Bash vous permet d'utiliser les flèches de déplacement pour insérer un caractère ou passer en revue l'historique. A cause de l'interactivité de ces commande, elles ne seront que rarement utiles pour les applications Web, mais peuvent se révéler utiles lorsqu'un script est exécuté depuis une ligne de commande .

Note

Cette extension n'est pas disponible sur les plates-formes Windows.

8.111.2 Pré-requis

Pour utiliser les fonctions readline, vous devez installer la bibliothèque libreadline. Vous pouvez la trouver sur la page du projet GNU, à <http://cnswww.cns.cwru.edu/~chet/readline/rktop.html> . Elle est maintenue par Chet Ramey, qui est l'auteur de Bash.

Vous pouvez aussi utiliser ces fonctions avec la bibliothèque libedit, un remplacement de la bibliothèque readline, qui n'est pas GPL. La bibliothèque libedit est sous license BSD, et elle est disponible depuis <http://sourceforge.net/projects/libedit/> .

8.111.3 Installation

Pour utiliser ces fonctions, vous devez compiler PHP, en version CGI ou CLI. Vous devez aussi utiliser l'option de compilation `--with-readline[=DIR]` . Si vous souhaitez utiliser la bibliothèque libedit readline, en remplacement, compilez PHP avec l'option `--with-libedit[=DIR]` .

8.111.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.111.5 Types de ressources

Cette extension ne définit aucune ressource.

8.111.6 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)

- [readline_add_history](#)
- [readline_callback_handler_install](#)
- [readline_callback_handler_remove](#)
- [readline_callback_read_char](#)
- [readline_clear_history](#)
- [readline_completion_function](#)
- [readline_info](#)
- [readline_list_history](#)
- [readline_on_new_line](#)
- [readline_read_history](#)
- [readline_redisplay](#)
- [readline_write_history](#)
- [readline](#)

8.111.8 [readline_callback_handler_install\(\)](#) : Initialise l'interface et le terminal de callback de readline, affiche le prompt et retourne immédiatement

`bool readline_callback_handler_install (string prompt , callback callback)`

[readline_callback_handler_install](#) définit une interface de callback pour readline, affiche le prompt et retourne immédiatement. La fonction callback prend un paramètre : l'entrée utilisateur retournée. Appeler cette fonction deux fois sans effacer au préalable l'interface de callback précédente effacera automatiquement et proprement l'ancienne interface.

La fonctionnalité de callback est très utile lorsque combinée à la fonction [stream_select](#) permettant l'interconnexion IO / entrée utilisateur, à la différence de [readline](#) .

8.111.9 [readline_callback_handler_remove\(\)](#) : Efface un gestionnaire de callback installé précédemment et restaure les paramètres du terminal

`bool readline_callback_handler_remove (void)`

[readline_callback_handler_remove](#) efface un gestionnaire de callback installé précédemment et restaure les paramètres du terminal.

8.111.10 [readline_callback_read_char\(\)](#) : Lit un caractère et informe l'interface de callback readline lorsqu'une ligne est reçue

`void readline_callback_read_char ()`

[readline_callback_read_char](#) lit un caractère de l'entrée utilisateur. Lorsqu'une ligne est reçue, la fonction informe l'interface de callback readline installée en utilisant [readline_callback_handler_install](#) qu'une ligne est prête à être entrée.

8.111.11 [readline_clear_history\(\)](#) : Efface l'historique

`bool readline_clear_history (void)`

readline_clear_history efface tout l'historique.

8.111.12 readline_completion_function() : Enregistre une fonction de complétion

bool readline_completion_function (callback function)

readline_completion_function enregistre une nouvelle fonction de complétion. Vous devez fournir le nom d'une fonction qui accepte un nom partiel de commande, et retourne une liste de fonctions complète possibles. C'est la même fonctionnalité que lorsque vous utilisez la touche de tabulation sous Bash.

8.111.13 readline_info() : Lit/modifie diverses variables internes

mixed readline_info (string varname , string newvalue)

Appelée sans paramètre, readline_info retourne un tableau contenant les valeurs des paramètres de Readline. Les éléments seront indexés par les clés suivantes : done, end, erase_empty_line, library_version, line_buffer, mark, pending_input, point, prompt, readline_name, et terminal_name.

Appelée avec le paramètre varname , la valeur de cette variable sera retournée. Appelée avec deux paramètres, et la valeur de la variable varname , sera remplacée par newvalue .

8.111.14 readline_list_history() : Liste l'historique

array readline_list_history (void)

readline_list_history retourne un tableau avec la liste de toutes les lignes de commandes de l'historique. Les éléments sont indexés numériquement, à partir de 0.

8.111.15 readline_on_new_line() : Informe readline que le curseur est passé à une nouvelle ligne

void readline_on_new_line (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.111.16 readline_read_history() : Lit l'historique

bool readline_read_history (string filename)

readline_read_history lit une ligne de l'historique depuis le fichier filename .

8.111.17 readline_redisplay() : Demande à readline de refaire l'affichage

void **readline_redisplay** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.111.18 readline_write_history() : Ecrit dans l'historique

bool **readline_write_history** (*string filename*)

readline_write_history écrit l'historique dans le fichier filename .

8.111.19 readline() : Lit une ligne

string **readline** (string prompt)

readline retourne une ligne entrée par l'utilisateur. Vous pouvez spécifier une chaîne de prompt. La ligne retournée est débarrassée du caractère nouvelle ligne final. Vous devez ajouter cette ligne à l'historique vous-même, avec la fonction readline_add_history .

Exemple avec readline

```
<?php
//Lit 3 commandes de l'utilisateur
for ($i=0; $i < 3; $i++) {
    $line = readline("Commande : ");
    readline_add_history($line);
}

//Liste l'historique
print_r(readline_list_history());

//liste les variables
print_r(readline_info());

?>
```

8.112 Recode (GNU)

8.112.1 Introduction

Ce module contient l'interface à la bibliothèque GNU Recode library, version 3.5. La bibliothèque GNU Recode library convertit les fichiers ayant des jeux de caractères différents. Lorsque ce n'est pas possible, elle se débarrasse des caractères illégaux, ou bien effectue une approximation. La bibliothèque reconnaît ou produit près de 150 jeux de caractères différents, et peut quasiment tous les convertir de l'un vers l'autre. La plupart des jeux de caractères de la [RFC 1345](#) sont supportés.

Note

Cette extension n'est pas disponible sur les plates-formes Windows.

8.112.2 Pré-requis

il faut que vous ayez la bibliothèque GNU Recode 3.5 ou plus récente, installée sur votre système. Vous pouvez télécharger GNU Recode [ici](#) .

Attention

La bibliothèque GNU Recode 3.6 ajoute des caractères étranges à la fin de la chaîne recodée dans certaines circonstances. Il est plus prudent d'utiliser la version 3.5 de la bibliothèque GNU Recode ou d'utiliser une des alternatives comme les extensions iconv ou mbstring .

8.112.3 Installation

Pour utiliser ces fonctions, vous devez compiler PHP avec l'option `--with-recode[=DIR]` .

Attention

Des crashes et des problèmes de démarrage de PHP peuvent être rencontrés lorsque l'extension recode est chargée APRES les extensions MySQL ou imap . Charger l'extension recode avant ces deux extensions corrige le problème. Ceci est dû à un problème technique car la bibliothèque c-client de IMAP et recode ont toutes les deux leur propre fonction <code>hash_lookup()</code> et les extensions <code>mysql</code> et <code>recode</code> ont toutes les deux leur fonction <code>hash_insert</code> .

Attention

L'extension IMAP ne peut être utilisée simultanément avec l'extension recode , l'extension YAZ ou l'extension Cyrus . Ceci est dû au fait que ces bibliothèques utilisent un symbole interne commun.
--

8.112.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.112.5 Types de ressources

Cette extension ne définit aucune ressource.

8.112.6 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [recode_file](#)
- [recode_string](#)
- [recode](#)

8.112.8 `recode_string()` : Recode une chaîne en fonction de la requête

`string recode_string (string request , string string)`

`recode_string` recode la chaîne `string` en fonction de la requête `request` . `recode_string` retourne la chaîne recodée en cas de succès et `FALSE` sinon.

Une requête simple de recodage peut être "lat1..iso646-de". Reportez-vous à la documentation GNU Recode de votre installation pour plus de détails sur les requêtes.

Exemple simple avec `recode_string`

```
<?php
echo recode_string("us..flat", "Le caractère suivant est diacritique : &acute;");
?>
```

8.112.9 `recode()` : Alias de `recode_string`

Cette fonction est un alias de : `recode_string` .

8.113 Expressions rationnelles

8.113.1 Introduction

Array

Attention

Ces expressions rationnelles ne sont pas compatibles avec les données binaires. Les expressions PCRE le sont.

Les expressions rationnelles sont utilisées pour de complexes manipulations sur les chaînes de caractères. PHP utilise les expressions rationnelles avancées de POSIX (POSIX 1003.2). Pour avoir tous les détails sur ces expressions, reportez-vous aux pages de manuel incluses dans le répertoire de la distribution PHP. Elles sont au format manpage : pour les lire, vous devrez utiliser la ligne de commande `man /usr/local/src/regex/regex.7` .

8.113.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.113.3 Installation

Attention

Ne changez pas la valeur de TYPE à moins que vous ne sachiez ce que vous faites.

Pour activer le support des expressions rationnelles en PHP vous devez compiler PHP avec l'option `--with-regex[=TYPE]` . TYPE peut prendre l'une des valeurs suivantes : "system", "apache", "php". La valeur par défaut est "php".

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de bibliothèque supplémentaire pour disposer de ces fonctions.

8.113.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.113.5 Types de ressources

Cette extension ne définit aucune ressource.

8.113.6 Constantes pré-définies

Cette extension ne définit aucune constante.

8.113.7 Exemples

Expressions rationnelles

```

<?php
/* Retourne
<font face="courrier">TRUE</font>
si "abc"
   est trouvé quelque part dans la chaîne $string. */
ereg ("abc", $string);

/* Retourne
<font face="courrier">TRUE</font>
si "abc"
   est trouvé au début de la chaîne $string. */
ereg ("^abc", $string);

/* Retourne
<font face="courrier">TRUE</font>
si "abc"
   est trouvé à la fin de la chaîne $string. */
ereg ("abc$", $string);

/* Retourne
<font face="courrier">TRUE</font>
si le client
   est Netscape 2, 3 ou MSIE 3. */
eregi ("(ozilla.[23]|MSIE.3)", $HTTP_USER_AGENT);

/* Recherche trois mots séparés par des espaces
   dans les chaînes $regs[1], $regs[2] et $regs[3]. */
ereg ("([[:alnum:]]+) ([[:alnum:]]+) ([[:alnum:]]+)", $string,$regs);

/* Insère une balise <br /> au début de la chaîne $string. */
$string = ereg_replace ("^", "<br />", $string);

/* Insère une balise <br /> à la fin de la chaîne $string. */
$string = ereg_replace ("$", "<br />", $string);

/* Supprime toutes les nouvelles lignes de $string. */
$string = ereg_replace ("\n", "", $string);

?>

```

8.113.8 Voir aussi

Pour des expressions rationnelles compatibles Perl, voyez l'extension [PCRE](#) . Les commandes de recherche plus simples (avec support des caractères jokers) sont fournies par la fonction [fnmatch](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [Voir aussi](#)
- [ereg_replace](#)
- [ereg](#)
- [eregi_replace](#)
- [eregi](#)

- [split](#)
- [spliti](#)
- [sql_regcase](#)

8.113.10 [ereg\(\)](#) : Expression rationnelle standard

int [ereg](#) (string pattern , string string , array regs)

Note

[preg_replace](#) , qui utilise la syntaxe des expressions rationnelles compatibles PERL, est une alternative plus rapide de [ereg_replace](#) .

[preg_replace](#) recherche dans la chaîne string les séquences de caractères qui correspondent au masque pattern , en tenant compte de la casse.

Si au moins une séquence est trouvée (éventuellement dans les parenthèses capturantes de pattern), et que la fonction est appelée avec un troisième argument regs , les résultats seront enregistrés dans regs . \$regs[1] contiendra la première parenthèse capturante (celle qui commence le plus tôt), \$regs[2] contiendra la deuxième parenthèse capturante (celle qui commence après la première), et ainsi de suite. \$regs[0] contient une copie de la chaîne.

Note

Pour les version de PHP supérieures ou égales à 4.1.0, \$regs sera rempli d'exactly dix éléments, même si il y a plus ou moins de dix parenthèses capturantes. Cela n'a aucun effet sur [ereg](#) pour chercher plus de sous-chaînes. Si aucun résultat n'est trouvé, \$regs ne sera pas altéré par [ereg](#) .

[ereg](#) retourne la longueur de l'occurrence trouvée si une occurrence a été trouvée dans la chaîne string et FALSE dans le cas contraire, ou si une erreur est survenue.

L'exemple suivant prend une date au format ISO (YYYY-MM-DD) et l'affiche sous la forme DD.MM.YYYY :

Exemple avec [ereg](#)

```
<?php
if (ereg ("([0-9]{4})-([0-9]{1,2})-([0-9]{1,2})", $date, $regs)) {
    echo "$regs[3].$regs[2].$regs[1]";
} else {
    echo "Format de date invalide : $date";
}
?>
```

Voir aussi [eregi](#) , [preg_match](#) , [ereg_replace](#) , [eregi_replace](#) , [strpos](#) et [strstr](#) .

8.113.11 [eregi_replace\(\)](#) : Remplacement par expression rationnelle insensible à la casse

string [eregi_replace](#) (string pattern , string replacement , string string)

[eregi_replace](#) est identique à [ereg_replace](#) , hormis le fait qu'elle ne tient pas compte de la casse des caractères alphabétiques.

Mise en évidence des résultats de la recherche

```
<?php
$pattern = '(>[^<]*)(' . quotemeta($_GET['search']) . ')';
$replacement = '\\1<span class="search">\\2</span>';
$body = eregi_replace($pattern, $replacement, $body);
?>
```

Voir aussi [ereg](#) , [eregi](#) et [ereg_replace](#) .

8.113.12 eregi() : Recherche par expression rationnelle insensible à la casse

int **eregi** (string pattern , string string , array regs)

[ereg](#) est identique à [ereg](#) , hormis le fait qu'elle ignore la casse des caractères lors de la recherche sur les caractères alphabétiques.

Exemple avec [eregi](#)

```
<?php
$string = 'XYZ';
if (eregi('z', $string)) {
    echo "'$string' contient un 'z' ou un 'Z!";
}
?>
```

Voir aussi [ereg](#) , [ereg_replace](#) , [eregi_replace](#) , [stripes](#) et [stristr](#) .

8.113.13 split() : Scinde une chaîne en un tableau, grâce à une expression rationnelle

array **split** (string pattern , string string , int limit) Array

[split](#) retourne un tableau de chaînes : chacune d'entre elle est une sous-chaîne de string délimitée par les occurrences trouvées de l'expression rationnelle pattern . Si limit est défini, le tableau retourné contiendra un maximum de limit éléments avec le dernier élément contenant le reste de la chaîne string . Si une erreur survient, retourne FALSE .

Pour lire les 5 premiers champs d'une ligne du fichier /etc/passwd :

Exemple avec [split](#)

```
<?php
list($user, $pass, $uid, $gid, $extra) = split (":", $passwd_line, 5);
?>
```

S'il y a n occurrences de pattern , le tableau retourné contiendra n +1 éléments. Par exemple, s'il n'y a aucune occurrence de pattern , un tableau d'un seul élément sera retourné. Bien sûr, cela reste vrai si string est une chaîne vide.

Pour analyser une date qui est délimitée par des /, des points ou des tirets :

Exemple avec [split](#)

```
<?php
// Les délimiteurs peuvent être des tirets, points ou slash
$date = "04/30/1973";
list($month, $day, $year) = split('[/.-]', $date);
```

```
echo "Month: $month; Day: $day; Year: $year<br />\n";
?>
```

Pour les utilisateurs qui recherchent le moyen d'émuler la commande Perl `@chars = split(", $str)`, voyez [preg_split](#).

Notez aussi que `pattern` est une expression rationnelle. Si vous voulez utiliser n'importe quel caractère spécial des expressions rationnelles, vous devez les échapper. Si vous pensez que `split` (ou toute autre expression rationnelle) se comporte bizarrement, lisez d'abord le fichier `regex.7`, inclus dans le dossier `regex/` de la distribution PHP. Il est au format `manpage`, et vous pourrez le lire avec une commande telle que `man /usr/local/src/regex/regex.7`.

Voir aussi [preg_split](#), [spliti](#), [explode](#), [implode](#), [chunk_split](#) et [wordwrap](#).

8.113.14 `spliti()` : Scinde une chaîne en un tableau, grâce à une expression rationnelle

array `spliti` (string `pattern`, string `string`, int `limit`)

`spliti` est identique à `split`, hormis le fait qu'elle ignore la casse.

Scinder une chaîne en utilisant la lettre 'a' comme séparateur :

Exemple avec `spliti`

```
<?php
$string = "aBBBaCCCADDaEEeaGGGA";
$chunks = spliti ("a", $string, 5);
print_r($chunks);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[0] =>
[1] => BBB
[2] => CCC
[3] => DDD
[4] => EEeaGGGA
)
```

Voir aussi [preg_split](#), [split](#), [explode](#) et [implode](#).

8.113.15 `sql_regcase()` : Prépare une expression rationnelle pour effectuer une recherche insensible à la casse

string `sql_regcase` (string `string`)

`sql_regcase` retourne une expression rationnelle valide qui acceptera la chaîne `string`, et toutes les variantes majuscule/minuscule possibles de cette chaîne. Cette expression sera construite à partir de la chaîne `string` en remplaçant tous les caractères par des expressions entre crochets (des

classes de caractères), contenant la lettre majuscule et minuscule. Si le caractère n'est pas une lettre, les crochets contiendront deux fois le caractère original.

Exemple avec `sql_regcase`

```
<?php
echo sql_regcase ("Foo - bar.");
?>
```

L'exemple ci-dessus va afficher :

`[Ff][Oo][Oo] - [Bb][Aa][Rr]`.

Cette expression sert à effectuer des recherches insensibles à la casse avec d'autres logiciels, qui n'acceptent que les recherches sensibles à la casse.

8.114 Fonctions runkit

8.114.1 Introduction

L'extension runkit fournit les moyens de modifier les constantes, les fonctions et les classes définies par l'utilisateur. Elle fournit aussi ces moyens pour les variables superglobales et les sous-interpréteurs par sandboxing.

Des informations comme les notes sur les nouvelles versions, les téléchargements, les sources des fichiers, les informations concernant les mainteneurs ainsi qu'un CHANGELOG, peuvent être trouvées ici : <http://pecl.php.net/package/runkit> .

Ce paquetage est signifié en tant que remplacement de fonctionnalités pour le paquetage `classkit` . Lorsque compilé avec l'option `--enable-runkit=classkit` à `./configure`, les définitions des fonctions et des constantes de `classkit` seront exportées.

8.114.2 Pré-requis

Modification de Constantes, Fonctions, Classes et Méthodes fonctionne avec toutes les versions de PHP 4 et PHP 5. Aucune condition spéciale n'est nécessaire.

Les Superglobales Personnalisées sont seulement disponibles dans les versions PHP 4.2.0 et supérieures.

Sandboxing nécessite PHP 5.1.0 ou supérieur ou PHP 5.0.0 avec le patch spécial TSRM appliqué. Sans se soucier de quelle version de PHP est utilisé, sandboxing doit être compilé avec l'option `--enable-maintainer-zts` . Voyez le fichier README dans le paquetage de runkit pour plus d'informations.

8.114.3 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable	Historique
<code>runkit.superglobal</code>	""	<code>PHP_INI_PERDIR</code>	
<code>runkit.internal_override</code>	"0"	<code>PHP_INI_SYSTEM</code>	

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

`runkit.superglobal` string

Liste de noms de variable séparés par des virgules à être traités comme variables superglobales. Cette valeur devrait être fixée dans le fichier du système `php.ini` , mais devrait fonctionner dans les contextes de configuration par dossier dépendamment de votre SAPI.

Superglobales particulières avec `runkit.superglobal=_FOO,_BAR` dans `php.ini`

```
<?php
function montre_valeurs() {
    echo "Foo est $_FOO\n";
    echo "Bar est $_BAR\n";
}
```

```

    echo "Baz est $_BAZ\n";
}

$_FOO = 'foo';
$_BAR = 'bar';
$_BAZ = 'baz';

/* Montre foo et bar mais pas baz */
montre_valeurs();
?>

```

8.114.4 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

RUNKIT_IMPORT_FUNCTIONS ([entier](#))

[runkit_import](#) marque indiquant que les fonctions normales doivent être importées à partir du fichier spécifié.

RUNKIT_IMPORT_CLASS_METHODS ([entier](#))

[runkit_import](#) marque indiquant que les méthodes de classes doivent être importées à partir du fichier spécifié.

RUNKIT_IMPORT_CLASS_CONSTS ([entier](#))

[runkit_import](#) marque indiquant que les constantes de classes doivent être importées à partir du fichier spécifié. Notez que cette marque est seulement signifiante dans les versions PHP 5.1.0 et supérieures.

RUNKIT_IMPORT_CLASS_PROPS ([entier](#))

[runkit_import](#) marque indiquant que les propriétés standards de classes doivent être importée à partir du fichier spécifié.

RUNKIT_IMPORT_CLASSES ([entier](#))

[runkit_import](#) marque indiquant une opération de bits OU (OR) des constantes `RUNKIT_IMPORT_CLASS_*`.

RUNKIT_IMPORT_OVERRIDE ([entier](#))

[runkit_import](#) marque indiquant que si n'importe quelles fonctions, méthodes, constantes ou propriétés importées existent, elles doivent être remplacées par leurs nouvelles définitions. Si ce drapeau n'est pas activé, alors toutes définitions importées qui existent déjà seront supprimées.

RUNKIT_ACC_PUBLIC ([entier](#))

Drapeau spécifique à PHP5 pour [runkit_method_add](#)

RUNKIT_ACC_PROTECTED ([entier](#))

Drapeau spécifique à PHP5 pour [runkit_method_add](#)

RUNKIT_ACC_PRIVATE ([entier](#))

Drapeau spécifique à PHP5 pour [runkit_method_add](#)

CLASSKIT_ACC_PUBLIC ([entier](#))

Drapeau spécifique à PHP5 pour [runkit_method_add](#) Seulement définie lorsque les compatibilités classkit sont activées.

CLASSKIT_ACC_PROTECTED ([entier](#))

Drapeau spécifique à PHP5 pour [runkit_method_add](#) Seulement définie lorsque les compatibilités classkit sont activées.

CLASSKIT_ACC_PRIVATE ([entier](#))

Drapeau spécifique à PHP5 pour [runkit_method_add](#) Seulement définie lorsque les compatibilités classkit sont activées.

CLASSKIT_AGGREGATE_OVERRIDE ([entier](#))

Drapeau spécifique à PHP5 pour `classkit_import` Seulement définie lorsque les compatibilités classkit sont activées.

`RUNKIT_VERSION` (chaîne de caractères)

Définie la version courante du paquetage runkit.

`CLASSKIT_VERSION` (chaîne de caractères)

Définie la version courante du paquetage runkit. Seulement définie lorsque les compatibilités classkit sont activées.

- [Introduction](#)
- [Pré-requis](#)
- [Configuration à l'exécution](#)
- [Constantes pré-définies](#)
- [Runkit_Sandbox](#)
- [Runkit_Sandbox_Parent](#)
- [runkit_class_adopt](#)
- [runkit_class_emancipate](#)
- [runkit_constant_add](#)
- [runkit_constant_redefine](#)
- [runkit_constant_remove](#)
- [runkit_function_add](#)
- [runkit_function_copy](#)
- [runkit_function_redefine](#)
- [runkit_function_remove](#)
- [runkit_function_rename](#)
- [runkit_import](#)
- [runkit_lint_file](#)
- [runkit_lint](#)
- [runkit_method_add](#)
- [runkit_method_copy](#)
- [runkit_method_redefine](#)
- [runkit_method_remove](#)
- [runkit_method_rename](#)
- [runkit_return_value_used](#)
- [runkit_sandbox_output_handler](#)
- [runkit_superglobals](#)

8.114.6 Runkit_Sandbox_Parent() : Classe Anti-Sandbox Runkit

void **Runkit_Sandbox_Parent::__construct** (void)

L'instanciation de la classe `Runkit_Sandbox_Parent` à l'intérieure d'un environnement sandbox créé à partir de la classe `Runkit_Sandbox` fournit certains moyens (contrôlés) pour un sandbox fils pour accéder à son parent.

Note

Support Sandbox (requis pour `runkit_lint` , `runkit_lint_file` et la classe `Runkit_Sandbox`) n'est seulement disponible qu'avec PHP 5.1 ou les versions de PHP 5.0 spécialement patché et nécessite que la protection de thread soit activée. Voyez le fichier README incluse dans le paquetage runkit pour plus d'informations.

Afin de faire fonctionner une des fonctionnalités de `Runkit_Sandbox_Parent` , le support doit être activé sur une base de chaque sandbox en activant le drapeau `parent_access` à partir du contexte parent.

Travailler avec des variables dans un sandbox

```
<?php
$sandbox = new Runkit_Sandbox();
$sandbox['parent_access'] = true;
?>
```

8.114.7 `runkit_class_adopt()` : Convertit une classe de base à une classe héritée, ajoute une méthode ancestrale lorsque approprié

```
bool runkit_class_adopt ( string classname , string parentname )
```

8.114.8 `runkit_class_emancipate()` : Convertit une classe héritée à une classe de base, supprime toute méthode pour qui la portée est ancestrale

```
bool runkit_class_emancipate ( string classname )
```

8.114.9 `runkit_constant_add()` : Similaire à `define()`, mais permet aussi la définition dans définitions de classe

```
bool runkit_constant_add ( string constname , mixed value )
```

8.114.10 `runkit_constant_redefine()` : Redéfinit une constante déjà définie

```
bool runkit_constant_redefine ( string constname , mixed newvalue )
```

8.114.11 `runkit_constant_remove()` : Enlève/Supprime une constante déjà définie

```
bool runkit_constant_remove ( string constname )
```

8.114.12 `runkit_function_add()` : Ajoute une nouvelle fonction, similaire à `create_function`

```
bool runkit_function_add ( string funcname , string arglist , string code )
```

8.114.13 `runkit_function_copy()` : Copie une fonction vers un nom de fonction nouveau

```
bool runkit_function_copy ( string funcname , string targetname )
```

8.114.14 `runkit_function_redefine()` : Remplace une définition de fonction avec une nouvelle implémentation

`bool runkit_function_redefine (string funcname , string arglist , string code)`

Note

Par défaut, seulement les fonctions définies par l'utilisateur peuvent être supprimées, renommées ou modifiées. Afin de surcharger des fonctions internes, vous devez activer la configuration `runkit.internal_override` dans le fichier `php.ini` du système entier.

8.114.15 `runkit_function_remove()` : Enlève une définition de fonction

`bool runkit_function_remove (string funcname)`

Note

Par défaut, seulement les fonctions définies par l'utilisateur peuvent être supprimées, renommées ou modifiées. Afin de surcharger des fonctions internes, vous devez activer la configuration `runkit.internal_override` dans le fichier `php.ini` du système entier.

8.114.16 `runkit_function_rename()` : Change le nom d'une fonction

`bool runkit_function_rename (string funcname , string newname)`

Note

Par défaut, seulement les fonctions définies par l'utilisateur peuvent être supprimées, renommées ou modifiées. Afin de surcharger des fonctions internes, vous devez activer la configuration `runkit.internal_override` dans le fichier `php.ini` du système entier.

8.114.17 `runkit_import()` : Traite un fichier PHP important fonctions et définitions de classes, écrasement où applicable

`bool runkit_import (string filename , int flags)`

Similaire à `include` , par contre tout code qui réside à l'extérieur de fonction ou classe est simplement ignoré. De plus, dépendamment de la valeur de `flags` , toutes fonctions et classes qui existent déjà dans l'environnement en cours d'exécution seront automatiquement écrasées par leurs nouvelles définitions.

8.114.18 `runkit_lint_file()` : Vérifie la syntaxe PHP d'un fichier spécifié

`bool runkit_lint_file (string filename)`

La fonction `runkit_lint_file` effectue une vérification de syntaxe (lint) sur le fichier spécifié en testant les erreurs de scripts. Cette fonction est similaire à l'utilisation de `php -l` à partir de la ligne de commande.

Note

Support Sandbox (requis pour `runkit_lint` , `runkit_lint_file` et la classe `Runkit_Sandbox`) n'est seulement disponible qu'avec PHP 5.1 ou les versions de PHP 5.0 spécialement patché et nécessite que la protection de thread soit activée. Voyez le fichier README incluse dans le

paquetage <code>runkit</code> pour plus d'informations.

8.114.19 `runkit_lint()` : Vérifie la syntaxe PHP de code PHP spécifié

`bool runkit_lint (string code)`

La fonction `runkit_lint` effectue une vérification de syntaxe (lint) sur le code PHP spécifié en testant les erreurs de scripts. Cette fonction est similaire à l'utilisation de `php -l` à partir de la ligne de commande à l'exception que `runkit_lint` accepte le code actuel plutôt que d'un fichier.

Note

Support Sandbox (requis pour <code>runkit_lint</code> , <code>runkit_lint_file</code> et la classe <code>Runkit_Sandbox</code>) n'est seulement disponible qu'avec PHP 5.1 ou les versions de PHP 5.0 spécialement patché et nécessite que la protection de thread soit activée. Voyez le fichier README incluse dans le paquetage <code>runkit</code> pour plus d'informations.

8.114.20 `runkit_method_add()` : Ajoute dynamiquement une nouvelle méthode à une classe donnée

`bool runkit_method_add (string classname , string methodname , string args , string code , int flags)`

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

8.114.21 `runkit_method_copy()` : Copie une méthode d'une classe à une autre

`bool runkit_method_copy (string dClass , string dMethod , string sClass , string sMethod)`

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

8.114.22 `runkit_method_redefine()` : Change dynamiquement le code de la méthode donnée

`bool runkit_method_redefine (string classname , string methodname , string args , string code , int flags)`

Note

Cette fonction ne peut être utilisée pour manipuler la méthode en cours d'utilisation (ou chaînée).

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

8.114.23 `runkit_method_remove()` : Supprime dynamiquement la méthode donnée

`bool runkit_method_remove (string classname , string methodname)`

Note
Cette fonction ne peut être utilisée pour manipuler la méthode en cours d'utilisation (ou chaînée).
Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

8.114.24 `runkit_method_rename()` : Change dynamiquement le nom de la méthode donnée

`bool runkit_method_rename (string classname , string methodname , string newname)`

Note
Cette fonction ne peut être utilisée pour manipuler la méthode en cours d'utilisation (ou chaînée).
Attention
Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

8.114.25 `runkit_return_value_used()` : Détermine si la valeur de retour des fonctions courantes sera utilisée

`bool runkit_return_value_used (void)`

8.114.26 `runkit_sandbox_output_handler()` : Spécifie une fonction à capturer et/ou traiter la sortie à partir d'un runkit sandbox

`mixed runkit_sandbox_output_handler (object sandbox , mixed callback)`

Normalement, toutes les sorties (comme avec `echo` ou `print`) seront écrites comme si elles avaient été écrites à partir de la portée du parent. Cependant, l'utilisation de `runkit_sandbox_output_handler` , sorties générées par le sandbox (incluant les erreurs), peuvent être capturées par une fonction extérieure à sandbox.

Note
Support Sandbox (requis pour <code>runkit_lint</code> , <code>runkit_lint_file</code> et la classe <code>Runkit_Sandbox</code>) n'est seulement disponible qu'avec PHP 5.1 ou les versions de PHP 5.0 spécialement patché et nécessite que la protection de thread soit activée. Voyez le fichier README incluse dans le paquetage <code>runkit</code> pour plus d'informations.
Note
Dépréciée
Depuis la version de <code>runkit</code> 0.5, cette fonction est dépréciée et devrait être supprimée de ce paquetage avant la version 1.0. Le gestionnaire de sortie pour une instance donnée de

`Runkit_Sandbox` devrait être lue/fixée en utilisant la syntaxe tableau de décalage montrée sur la page de définition de la classe `Runkit_Sandbox` .

8.114.27 `runkit_superglobals()` : Retourne un tableau indexé numériquement des variables superglobales enregistrées

array `runkit_superglobals` (void)

8.115 Sémaphores et gestion de la mémoire partagée

8.115.1 Introduction

Ce module fournit un système de sémaphores. Ce système utilise les sémaphores System V.

Les sémaphores peuvent être utilisés pour fournir un accès exclusif à certaines ressources de la machine, ou pour limiter le nombre de processus qui utilisent en même temps une ressource.

Ce module fournit aussi un système de mémoire partagée, qui utilise la mémoire partagée System V. Cette mémoire partagée permet d'accéder à des variables globales. Les différents démons httpd et même d'autres programmes (tels que Perl, C...) permettent un tel échange de données global. N'oubliez pas que la mémoire partagée n'est pas protégée contre l'accès simultané. Il vous faudra utiliser les sémaphores pour assurer la synchronisation.

SHMMAX	Taille maximale de mémoire partagée, par défaut, 131072 octets.
SHMMIN	Taille minimale de mémoire partagée, par défaut, 1 octet.
SHMMNI	Nombre maximal de segment de mémoire partagé, par défaut 100.
SHMSEG	Taille maximale de mémoire partagée par processus, par défaut 6.

Ces fonctions permettent d'envoyer et de recevoir des messages de/vers d'autres processus. Elles offrent une interface simple et efficace pour échanger des données entre processus, sans avoir besoin d'utiliser un autre socket UNIX.

Note
Cette extension n'est pas disponible sur les plates-formes Windows.

8.115.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.115.3 Installation

Le support de ces fonctions n'est pas activé par défaut. Pour activer le support des sémaphores System V, compilez PHP avec l'option `--enable-sysvsem`. Pour activer le support de la mémoire partagée System V, compilez PHP avec l'option `--enable-sysvshm`. Pour activer le support des messages System V, compilez PHP avec l'option `--enable-sysvmsg`.

8.115.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable	Historique
<code>sysvmsg.value</code>	"42"	PHP_INI_ALL	
<code>sysvmsg.string</code>	"foobar"	PHP_INI_ALL	

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#).

8.115.5 Types de ressources

8.115.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Constante	Type
MSG_IPC_NOWAIT	entier
MSG_NOERROR	entier
MSG_EXCEPT	entier

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [ftok](#)
- [msg_get_queue](#)
- [msg_receive](#)
- [msg_remove_queue](#)
- [msg_send](#)
- [msg_set_queue](#)
- [msg_stat_queue](#)
- [sem_acquire](#)
- [sem_get](#)
- [sem_release](#)
- [sem_remove](#)
- [shm_attach](#)
- [shm_detach](#)
- [shm_get_var](#)
- [shm_put_var](#)
- [shm_remove_var](#)
- [shm_remove](#)

8.115.8 [msg_get_queue\(\)](#) : Crée ou s'attache à une file de messages

resource [msg_get_queue](#) (int *key* , int *perms*)

[msg_get_queue](#) retourne une ressource qui peut être utilisée avec les files de messages System V message, et la clé *key* . Le premier appel à la fonction va créer la file de messages, avec les droits optionnels de *perms* (par défaut : 0666). Un second appel à [msg_get_queue](#) avec la même clé *key* retournera une autre ressource de file de messages, mais les deux identifiants aboutissent à la même file de messages. Si la file de messages existe déjà, le paramètre *perms* sera ignoré.

Voir aussi [msg_remove_queue](#) , [msg_receive](#) , [msg_send](#) , [msg_stat_queue](#) et [msg_set_queue](#) .

8.115.9 `msg_receive()` : Reçoit un message depuis une file de messages

`bool msg_receive (resource queue , int desiredmsgtype , int msgtype , int maxsize , mixed message , bool unserialize , int flags , int errorcode)`

`msg_receive` reçoit le premier message de la file `queue` , type spécifié par `desiredmsgtype` . Le type de message qui a été reçu sera stocké dans le paramètre `msgtype` . La taille maximale de message est fixée par `maxsize` ; si le message de la file est plus grand que cette taille, la fonction échouera (à moins que vous n'utilisiez une option `flags` , décrite ci-dessous). Le message reçu sera stocké dans le paramètre `message` , à moins qu'il n'y ait eu des erreurs de réception du message, auquel cas le paramètre optionnel `errorcode` se verra assigner le numéro d'erreur.

Si `desiredmsgtype` vaut 0, le premier message de la file est retourné. Si `desiredmsgtype` vaut plus que 0, alors le premier message de ce type sera retourné. Si `desiredmsgtype` vaut moins que 0, le premier message de la file ayant un type inférieur ou égal à la valeur absolue de `desiredmsgtype` sera retourné. Si aucun message ne correspond aux critères, votre script attendra la venue d'un tel message dans la file. Vous pouvez éviter ce blocage en précisant l'option `MSG_IPC_NOWAIT` dans le paramètre `flags` .

`unserialize` vaut par défaut `TRUE` ; quand c'est le cas, le message est traité comme s'il avait été linéarisé avec le même mécanisme que le module de session. Le message sera alors délinéarisé, puis retourné au script. Cela vous permettra de recevoir facilement des tableaux ou des objets complexes dans votre script, émis par d'autres scripts PHP. Si `unserialize` vaut `FALSE` , le message sera retourné intact, et sans modifier les valeurs binaires.

Le paramètre `flags` permet de passer des options pour configurer les appels `msgrcv`. Par défaut, il vaut 0, mais vous pouvez spécifier une ou plusieurs options en les combinant avec l'opérateur OR).

<code>MSG_IPC_NOWAIT</code>	S'il n'y a pas de message du type <code>desiredmsgtype</code> , retourne immédiatement, et n'attend pas. La fonction échouera et retournera un entier correspondant à <code>ENOMSG</code> .
<code>MSG_EXCEPT</code>	En utilisant cette option en combinaison avec un type <code>desiredmsgtype</code> supérieur à 0, la fonction va lire le premier message qui n'est pas du type demandé par <code>desiredmsgtype</code> .
<code>MSG_NOERROR</code>	Si le message est plus grand que <code>maxsize</code> , cette option va tronquer le message à la taille de <code>maxsize</code> et ne signalera pas d'erreur.

Lors de la réception réussie d'un message, la file est mise à jour comme ceci : `msg_lpid` prend la valeur de l'identifiant de processus du processus appelant, `msg_qnum` est décrementé de 1 et `msg_rtime` prend la date et l'heure courante.

`msg_receive` retourne `TRUE` en cas de succès et `FALSE` en cas d'échec. Si la fonction échoue, le paramètre optionnel `errorcode` contiendra le numéro d'erreur rencontré.

Voir aussi `msg_remove_queue` , `msg_send` , `msg_stat_queue` et `msg_set_queue` .

8.115.10 `msg_remove_queue()` : Détruit une file de messages

`bool msg_remove_queue (resource queue)`

`msg_remove_queue` détruit la file de messages identifiée par `queue` . Utiliser uniquement cette fonction lorsque tous les processus ont fini de travailler dans la file de message, et que vous voulez libérer les ressources.

Voir aussi [msg_get_queue](#) , [msg_receive](#) , [msg_stat_queue](#) et [msg_set_queue](#) .

8.115.11 [msg_send\(\)](#) : Envoie un message dans une file

bool **msg_send** (resource queue , int msgtype , mixed message , bool *serialize* , bool *blocking* , int *errorcode*)

[msg_send](#) envoie le message message de type msgtype (qui DOIT être plus grand que 0) à la file de messages identifiée par queue .

Si le message est trop grand pour être stocké par la file, votre script attendra jusqu'à ce qu'un autre processus lise dans la file un message, et libère assez de place pour votre message. C'est le mode bloquant : vous pouvez éviter ce mode en utilisant le paramètre blocking avec la valeur FALSE : dans ce cas, [msg_send](#) retournera immédiatement FALSE si le message est trop gros pour la file. Il assignera alors au paramètre errorcode la valeur de EAGAIN, indiquant que vous devriez essayer d'envoyer à nouveau votre message, un peu plus tard.

Le paramètre optionnel serialize contrôle la méthode d'envoi du message message . serialize faut par défaut TRUE ce qui signifie que le message message sera linéarisé en utilisant le même mécanisme que celui qui est utilisé par les sessions, avant d'être envoyé à la file de message. Cela permet d'envoyer des tableaux et des objets complexes à d'autres scripts PHP, ou bien, si vous utilisez l'extension WDDX, d'échanger des messages avec des clients compatibles WDDX.

Lors de l'émission réussie d'un message, la file est mise à jour comme ceci : msg_lrpuid prend la valeur de l'identifiant de processus du processus appelant, msg_qnum est décrementé de 1 et msg_rtime prend la date et l'heure courante.

Voir aussi [msg_remove_queue](#) , [msg_receive](#) , [msg_stat_queue](#) et [msg_set_queue](#) .

8.115.12 [msg_set_queue\(\)](#) : Modifie des informations dans la file de messages

bool **msg_set_queue** (resource queue , array data)

[msg_set_queue](#) vous permet de modifier certaines valeurs comme msg_perm.uid, msg_perm.gid, msg_perm.mode et msg_qbytes, qui sont des champs de la structure qui héberge la file de message. Il suffit de spécifier la valeur à modifier comme index dans le tableau data et la file queue sera modifiée.

Modifier la structure de données requiert que PHP fonctionne avec le même utilisateur que celui qui a créé la file, qui possède la file (comme déterminé par les champs msg_perm.xxx), ou qui fonctionne avec les droits de super utilisateur. Les droits de super utilisateur sont nécessaires pour affecter à msg_qbytes des valeurs supérieures aux limites du système.

Voir aussi [msg_remove_queue](#) , [msg_receive](#) , [msg_stat_queue](#) et [msg_get_queue](#) .

8.115.13 [msg_stat_queue\(\)](#) : Retourne des informations sur la file de messages

array **msg_stat_queue** (resource queue)

msg_stat_queue retourne des informations sur la file de messages identifiée par queue . C'est une fonction pratique pour connaître le processus qui a émis le message qui vient d'être reçu.

La valeur retournée par la fonction est un tableau dont les index et valeurs sont les suivants :

msg_perm.uid	L'uid du propriétaire de la file.
msg_perm.gid	Le gid du propriétaire de la file.
msg_perm.mode	Le mode d'accès à la file.
msg_stime	L'heure du dernier message envoyé à la file.
msg_rtime	L'heure du dernier message émis par la file.
msg_ctime	L'heure de modification de la file.
msg_qnum	Le nombre de messages en attente dans la file.
msg_qbytes	Le nombre d'octets actuellement disponibles dans la file pour stocker des messages jusqu'à leur lecture.
msg_lspid	Le pid du processus qui a envoyé le dernier message à la file.
msg_lrpid	Le pid du processus qui a reçu le dernier message de la file.

Voir aussi msg_remove_queue , msg_receive , msg_get_queue et msg_set_queue .

8.115.14 sem_acquire() : Réserve un sémaphore

bool **sem_acquire** (resource sem_identifieur)

sem_acquire se bloque (si nécessaire) jusqu'à ce que le sémaphore puisse être réservé. Un processus qui tente de réserver un sémaphore qu'il a déjà réservé restera en attente indéfinie, si cette acquisition excède le nombre max_acquire de réservation simultanée. sem_identifieur est une ressource de sémaphore, obtenue de la fonction sem_get .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

A la fin d'un script, tous les sémaphores réservés mais non explicitement libérés, seront libérés automatiquement, et une alerte sera générée.

Voir aussi sem_get et sem_release .

8.115.15 sem_get() : Retourne un identifiant de sémaphore

resource **sem_get** (int key , int max_acquire , int perm , int auto_release)

sem_get retourne un identifiant qui pourra être utilisé pour accéder à un sémaphore System V. Le sémaphore est créé avec l'identifiant key , si nécessaire, en utilisant les bits de droits du paramètre optionnel perm (par défaut, 0666). Le nombre de processus qui peuvent réserver simultanément le sémaphore est précisé dans le paramètre max_acquire (par défaut, 1). Actuellement, cette valeur n'est affectée que si le processus est le seul processus actuellement attaché au sémaphore.

Le paramètre optionnel auto_release spécifie si le sémaphore doit être automatiquement libéré à la fermeture. Il est disponible depuis PHP 4.3.0.

sem_get retourne une ressource de sémaphore en cas de succès, et FALSE en cas d'erreur.

Un deuxième appel à `sem_get` avec la même clé retournera un identifiant différent, mais les deux identifiants permettront d'accéder au même sémaphore.

Voir aussi `ftok` , `sem_acquire` et `sem_release` .

8.115.16 `sem_release()` : Libère un sémaphore

bool `sem_release` (resource sem_identifiant)

`sem_release` libère le sémaphore `sem_identifiant` , s'il a été réservé par le processus courant, sinon génère une erreur.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Après libération du sémaphore, `sem_acquire` peut être appelé pour le réserver à nouveau.

Voir aussi `sem_get` et `sem_acquire` .

8.115.17 `sem_remove()` : Détruit un sémaphore

bool `sem_remove` (resource sem_identifiant)

`sem_remove` supprime le sémaphore `sem_identifiant` s'il a été créé par `sem_get` et, sinon, émet une alerte.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Après suppression du sémaphore, il n'est plus utilisable.

Voir aussi `sem_get` , `sem_release` et `sem_acquire` .

8.115.18 `shm_attach()` : Crée ou ouvre un segment de mémoire partagée

int `shm_attach` (int key , int memsize , int perm)

`shm_attach` retourne un identifiant qui permettra d'accéder au System V de mémoire partagée. Au premier appel, la mémoire sera créée, avec la taille `memsize` (par défaut: `sysvshm.init_mem` dans `php.ini` , sinon 10000 octets) et avec les permissions `perm` (par défaut : 666).

Aux appels suivants avec la même clé `key` , `shm_attach` retournera un nouvel identifiant, mais cet identifiant accédera toujours à la même portion de mémoire partagée. Dans ce cas, `memsize` et `perm` seront ignorés.

Voir aussi `ftok` et `shm_detach` .

8.115.19 shm_detach() : Libère un segment de mémoire partagée

bool **shm_detach** (int shm_identifieur)

shm_detach libère le segment de mémoire partagée identifié par shm_identifieur et créé par shm_attach . N'oubliez pas que cette mémoire partagée existe toujours sous Unix, et que les données sont toujours accessibles.

shm_detach retourne toujours TRUE .

Voir aussi shm_attach , shm_remove et shm_remove_var .

8.115.20 shm_get_var() : Lit une variable dans la mémoire partagée

mixed **shm_get_var** (int shm_identifieur , int variable_key)

shm_get_var retourne la variable repérée par variable_key , dans le segment de mémoire partagée identifié par shm_identifieur . shm_identifieur est une ressource retournée par shm_attach . La variable est toujours présente en mémoire partagée.

8.115.21 shm_put_var() : Insère ou modifie une variable de la mémoire partagée

bool **shm_put_var** (int shm_identifieur , int variable_key , mixed variable)

shm_put_var insère ou modifie la variable variable avec la clé variable_key dans le segment de mémoire shm_identifieur . Tous les types de variables sont supportés.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Des alertes (niveau E_WARNING level) seront émises si shm_identifieur n'est pas un segment de mémoire système V valide, ou s'il n'y a pas assez de mémoire pour votre requête.

8.115.22 shm_remove_var() : Efface une variable de la mémoire partagée

bool **shm_remove_var** (int shm_identifieur , int variable_key)

shm_remove_var efface la variable variable_key de la mémoire partagée shm_identifieur et libère la mémoire.

Voir aussi shm_remove .

8.115.23 shm_remove() : Supprime un segment de mémoire partagée sous Unix

bool **shm_remove** (int shm_identifieur)

shm_remove supprime le segment de mémoire partagée shm_identifier . Toutes les données seront supprimées.

Voir aussi shm_remove_var .

8.116 SESAM

8.116.1 Introduction

SESAM/SQL-Server est une base de données mainframe, développée par Fujitsu Siemens Computers, Allemagne. Elle fonctionne sur les serveurs mainframe, sous BS2000/OSD.

Sur de nombreuses installation BS2000 en production, SESAM/SQL-Server a prouvé ...

- La facilité de connectivité Java, Web et client/serveur,
- La disponibilité de plus de 99.99%,
- La capacité de gérer des dizaines et mêmes des centaines de milliers d'utilisateurs.

Désormais, il existe une interface PHP pour SESAM, qui donne l'accès à cette base aux scripts PHP.

Note

L'accès aux bases SESAM n'est disponible que pour la dernière version CVS de PHP 3. PHP 4 ne supporte pas les bases SESAM.
--

8.116.2 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

`sesam_oml` string

Nom de la bibliothèque BS2000 PLAM contenant le module du pilote SESAM. Ceci est obligatoire pour utiliser les fonctions SESAM. La bibliothèque BS2000 PLAM doit être à ACCESS=READ,SHARE=YES car elle doit être lisible par le serveur Apache.

`sesam_configfile` string

Nom du fichier de configuration de l'application SESAM. Ceci est obligatoire pour utiliser les fonctions SESAM. Le fichier BS2000 doit être à ACCESS=READ,SHARE=YES car elle doit être lisible par le serveur Apache.

Le fichier de configuration de l'application contient généralement un fichier de configuration tel que celui-ci (reportez-vous au manuel de référence SESAM).

Configuration SESAM

CNF=B NAM=K NOTYPE

`sesam_messagecatalog` string

Nom du fichier de messages SESAM. Dans la plupart des cas, cette directive n'est pas nécessaire. Uniquement si le fichier de messages SESAM n'est pas installé dans la table de messages BS2000. Il peut alors être choisi avec cette directive.

Le catalogue de messages doit être à ACCESS=READ,SHARE=YES car elle doit être lisible par le serveur Apache.

8.116.3 Notes sur la configuration

Il n'y a pas de version exécutable indépendante pour l'interface PHP SESAM, et elle doit fonctionner en module Apache intégré. Dans le module PHP, l'interface SESAM est configurée avec des directives Apache.

Directive	Signification
php3_sesam_oml	Nom de la bibliothèque BS2000 PLAM qui contient les modules pilotes chargeables SESAM. Elle est nécessaire pour pouvoir utiliser les fonctions SESAM.
	Localisation du fichier BS2000 pour SESAM
	php3_sesam_oml \$.SYSLNK.SESAM-SQL.030
php3_sesam_configfile	Nom du fichier de configuration de l'application SESAM. Elle est nécessaire pour utiliser les fonctions SESAM.
	Configuration de php3_sesam_configfile
	php3_sesam_configfile \$SESAM.SESAM.CONF.AW
	Ce fichier va généralement contenir une configuration comme celle-ci. (reportez-vous au manuel de référence SESAM).
	Configuration SESAM
	CNF=B NAM=K NOTYPE
php3_sesam_messagecatalog	Nom du catalogue de messages SESAM. Dans la plupart des cas, cette directive n'est pas nécessaire. Uniquement dans les cas où le fichier de message n'est pas installé dans la table de message du système BS2000, vous pouvez utiliser cette directive :
	Nom du catalogue de messages SESAM
	php3_sesam_messagecatalog \$.SYSMES.SESAM-SQL.030

En plus de la configuration de l'interface PHP/SESAM, vous devez configurer le serveur SESAM-Database lui-même, sur votre mainframe habituel. Cela signifie :

- démarrer le gestionnaire de base SESAM (DBH)
- connecter les bases avec le gestionnaire de bases SESAM

Pour connecter un script PHP au serveur de bases SESAM, les paramètres CNF et NAM de la configuration SESAM sélectionnée doivent correspondre à l'id du gestionnaire de base démarré.

Dans le cas des bases de données distribuées, vous devez démarrer un agent SESAM/SQL-DCN, avec la table de distribution incluant le nom de l'hôte et de la base de données.

La communication entre PHP (fonctionnant sur le sous-système POSIX) et le gestionnaire de base (fonctionnant hors du sous-système POSIX) est réalisée par un pilote spécial appelé SQLSCI et le module de connexion SESAM, qui utilise la mémoire partagée. A cause de la mémoire partagée, et parce que PHP est une partie statique du serveur web, les accès à la base de données sont extrêmement rapides ; car ils ne requièrent pas de connexion distante via ODBC, JDBC ou UTM.

Seul un chargeur de stub (stub loader, SESMOD) est compilé dans PHP. Les modules de connexion SESAM proviennent de la bibliothèque OML PLAM. Dans la [configuration](#), vous devez indiquer à PHP le nom de la bibliothèque PALM, et le fichier de lien à utiliser pour la configuration de SESAM (En SESAM V3.0, SQLSCI est disponible dans la bibliothèque d'outils SESAM (SESAM Tool Library), qui fait partie de la distribution standard).

Les commandes SQL imposent que les guillemets simples soient doublés pour être interprétés littéralement (contrairement à d'autres bases de données qui utilisent un guillemet simple, précédé d'un antislash), il est recommandé d'activer les directives PHP [php3_magic_quotes_gpc](#) et [php3_magic_quotes_sybase](#).

8.116.4 Considérations à l'exécution

A cause des limitations du modèle de processus BS2000, le pilote peut être chargé uniquement après que le serveur Apache ait généré le processus fils. Cela ralentit légèrement le traitement de la première requête, mais toutes les requêtes suivantes seront effectuées à pleine vitesse.

Lorsque vous définissez explicitement le catalogue de messages SESAM, ce catalogue sera chargé à chaque fois que le pilote est chargé (i.e., au moment de la requête initiale). Le système d'exploitation BS2000 affiche un message après avoir correctement chargé le catalogue de messages, qui sera envoyé au fichier d'erreurs Apache. BS2000 ne permet pas la suppression de ce message, qui va remplir progressivement ce fichier.

Assurez-vous que la bibliothèque SESAM OML PLAM et le fichier de configuration SESAM sont accessibles par l'utilisateur qui fait tourner le serveur web. Sinon, le serveur ne sera pas capable de charger le pilote, ou d'appeler les fonctions SESAM. L'accès à la base doit être donné à cet utilisateur. Sinon, les connexions SESAM échoueront.

8.116.5 Types de curseurs

Les curseurs de résultats sont alloués pour les requêtes SQL de sélection ; ils peuvent être soit "séquentiels", soit "à défilement" ("scrollable"). Les curseurs à défilement sont beaucoup plus gourmands en mémoire, et le mode par défaut est séquentiel.

Lorsque vous utilisez les curseurs à défilement, le curseur peut être positionné librement dans le résultat. Pour chaque requête à défilement, il existe des valeurs globales de types de défilement (initialisée à : SESAM_SEEK_NEXT) et la position peut être fixée une seule fois par [sesam_seek_row](#) ou bien à chaque appel, avec la fonction [sesam_fetch_row](#). Lorsque vous lisez une ligne avec un curseur à défilement, le traitement suivant est effectué à partir des valeurs globales de type de défilement et de position :

Type de défilement	Action
SESAM_SEEK_NEXT	aucun
SESAM_SEEK_PRIOR	aucun
SESAM_SEEK_FIRST	le type de défilement devient SESAM_SEEK_NEXT
SESAM_SEEK_LAST	le type de défilement devient SESAM_SEEK_PRIOR
SESAM_SEEK_ABSOLUTE	incrémente automatiquement la valeur interne de position
SESAM_SEEK_RELATIVE	aucune. Conserve les valeurs globales par défaut de position, ce qui permet, par exemple, de lire toutes les 10 lignes, en arrière.

8.116.6 Portage

En PHP, il est naturel de commencer les index à zéro (plutôt que 1), et quelques adaptations ont été faites pour l'interface SESAM : à chaque fois qu'un tableau indexé commence à l'index 1 en SESAM natif, l'interface PHP utilisera l'index 0 comme point de départ. Par exemple, lorsque vous lisez des données avec `sesam_fetch_row`, la première colonne sera à l'index 0, et les suivantes suivront jusqu'au nombre de colonnes (exclus) du résultat (`$array["count"]`). Lors du portage d'applications depuis d'autres langages évolués vers PHP, soyez attentifs à ce changement. A chaque fois que c'est nécessaire, la description d'une fonction PHP SESAM indique que l'index du tableau commence à 0.

8.116.7 Sécurité

Lorsque vous autorisez l'accès à une base de données SESAM, le serveur web doit avoir le minimum de privilèges possible. Pour la plupart des bases de données, seul le droit de lecture doit être fourni. Suivant votre utilisation, ajoutez d'autres droits d'accès au fur et à mesure de vos besoins. Ne donnez jamais le contrôle total de vos bases à un utilisateur du web ! Limitez l'accès aux scripts PHP qui doivent administrer la base en utilisant un mot de passe et/ou une sécurisation SSL.

8.116.8 Migration d'une autre base SQL

Deux langages SQL ne sont jamais 100 % compatibles. Lorsque vous portez une application SQL depuis une autre interface vers SESAM, certaines adaptations doivent être faites. Les différences suivantes sont les plus courantes :

- Types de données spécifiques

Certains types de données spécifiques à une base doivent être remplacés par les types de données standard SQL. (i.e., TEXT doit être remplacé par VARCHAR(taille max)).

- Mots réservés comme identifiants SQL.

En SESAM (comme dans le standard SQL), les mots réservés utilisés comme identifiants doivent être entourés de guillemets doubles (ou renommés).

- Taille d'affichage des données.

Les types de données SESAM ont une taille de stockage, mais pas de taille d'affichage. A la place de `int(4)` (c'est-à-dire : les entiers jusqu'à '9999'), SESAM requiert simplement `int`, pour une taille implicite de 31 bits. De même, les seuls types de dates disponibles dans SESAM sont : `DATE`, `TIME(3)` et `TIMESTAMP(3)`.

- Les types de données unsigned (non signé), zerofill (complété avec des zéros), ou `auto_increment`

Unsigned et zerofill ne sont pas supportés. `Auto_increment` est automatique (utilisez `"INSERT ... VALUES(*, ...)"` au lieu de `"... VALUES(0,...)"` pour profiter des auto-increment implicites de SESAM.

- `int ... DEFAULT '0000'`

Les variables numériques ne doivent pas être initialisées avec des constantes de type chaîne de caractères. Utilisez `DEFAULT 0` à la place. Pour initialiser une date, la chaîne doit être préfixée avec le type de date adapté, tel que : `CREATE TABLE exmpl (xtime timestamp(3) DEFAULT TIMESTAMP '1970-01-01 00:00:00.000' NOT NULL);`

- \$count = xxxx_num_rows();

Certaines bases de données essaient d'estimer le nombre de lignes d'un résultat, même grossièrement approximativement. SESAM ne connaît pas le nombre de lignes avant de les avoir lues lui-même. Si vous avez vraiment besoin de les compter, utilisez la commande SELECT COUNT(...) WHERE ... , qui vous dira combien de lignes sont disponibles. Une deuxième requête devrait vous retourner tous ces résultats.

- DROP TABLE lenom;

Avec SESAM, dans la commande DROP TABLE , le nom de la table doit être suivi du mot-clé RESTRICT ou CASCADE . Avec RESTRICT , une erreur est retournée s'il y a des objets dépendants (par exemple, des vues), tandis qu'avec CASCADE , les objets dépendants seront supprimés en même temps que la table.

8.116.9 Notes sur l'utilisation de types SQL divers

SESAM ne supporte pas le type BLOB. Une future version de SESAM devra le faire.

L'interface PHP effectue automatiquement les conversions suivantes lors de la lecture de lignes de résultats SQL :

Type SQL	Type PHP
SMALLINT, INTEGER	"integer" (entier)
NUMERIC, DECIMAL, FLOAT, REAL, DOUBLE	"double" (nombre à virgule flottante)
DATE, TIME, TIMESTAMP	"string"(chaîne de caractères)
VARCHAR, CHARACTER	"string"(chaîne de caractères)

Lorsque vous lisez une ligne entière, le résultat est retourné sous la forme d'un tableau. Les champs vides ne sont pas remplis, et vous aurez à vérifier vous-même leur existence (utilisez isset ou empty pour tester les champs vides). Cela donne plus de contrôle à l'utilisateur sur l'apparence des champs que si les champs vides étaient représentés par des chaînes vides).

8.116.10 Support des "champs multiples" de SESAM

La fonctionnalité spéciale des "champs multiples" de SESAM permet à une colonne de contenir un tableau de champs. Un tel "champs multiples" peut être créé comme ceci :

Création d'une colonne de champs multiples
<pre>CREATE TABLE multi_field_test (pkey CHAR(20) PRIMARY KEY, multi(3) CHAR(12))</pre>

et peut être remplie avec :

Affectation d'une colonne de type "champs multiples"
<pre>INSERT INTO multi_field_test (pkey, multi(2..3)) VALUES ('Second', <'first_val', 'second_val'>)</pre>

Notez que (comme c'est le cas ci-dessus), les sous-champs vides initiaux sont ignorés, et que le tableau est alors compacté, ce qui fait que l'exemple ci-dessus conduit à un tableau multi(1..2) au lieu de multi(2..3).

Lors de la lecture d'une ligne, les "champs multiples" sont mis en colonnes. Dans l'exemple ci-dessus, "pkey" prend l'index 0, et les trois colonnes "multi(1..3)" sont accessibles depuis les index

1 à 3.

8.116.11 Voir aussi

Pour de plus amples détails sur SESAM, reportez-vous à la documentation [SESAM/SQL-Server](#) en anglais ou [SESAM/SQL-Server](#) en allemand, disponibles toutes deux en ligne, ou en manuels.

- [Introduction](#)
- [Configuration à l'exécution](#)
- [Notes sur la configuration](#)
- [Considérations à l'exécution](#)
- [Types de curseurs](#)
- [Portage](#)
- [Sécurité](#)
- [Migration d'une autre base SQL](#)
- [Notes sur l'utilisation de types SQL divers](#)
- [Support des "champs multiples" de SESAM](#)
- [Voir aussi](#)
- [sesam_affected_rows](#)
- [sesam_commit](#)
- [sesam_connect](#)
- [sesam_diagnostic](#)
- [sesam_disconnect](#)
- [sesam_errormsg](#)
- [sesam_execimm](#)
- [sesam_fetch_array](#)
- [sesam_fetch_result](#)
- [sesam_fetch_row](#)
- [sesam_field_array](#)
- [sesam_field_name](#)
- [sesam_free_result](#)
- [sesam_num_fields](#)
- [sesam_query](#)
- [sesam_rollback](#)
- [sesam_seek_row](#)
- [sesam_settransaction](#)

8.116.13 `sesam_commit()` : Valide la transaction SESAM en cours

bool `sesam_commit` (void)

`sesam_commit` retourne TRUE en cas de succès et FALSE sinon.

`sesam_commit` valide toutes les modifications de tables en attente sur la base.

Notez qu'il n'y a pas de mode "auto-commit", comme dans d'autres bases de données, car cela peut conduire à une perte accidentelle de données. Les données non valides à la fin d'un script (ou au moment de l'appel de `sesam_disconnect`) seront annulées par un appel implicite à `sesam_rollback` .

Valider une transaction SESAM

<?php

```

if(sesam_connect ("moncatalogue", "monschema", "toto")) {
    if(!sesam_execimm("INSERT INTO mytable VALUES (*, 'Small Test', <0, 8, 15>")) {
        die("Insertion manquée");
    }
    if (!sesam_commit()) {
        die("Insertion réussie");
    }
}
?>

```

Voir aussi [sesam_rollback](#) .

8.116.14 **sesam_connect()** : Ouvre une connexion SESAM

bool **sesam_connect** (string catalog , string schema , string user)

sesam_connect retourne TRUE si une connexion à la base SESAM a été faite, ou FALSE en cas d'erreur.

sesam_connect établit une connexion au serveur SESAM. La connexion est toujours "persistante", en ce sens que le pilote sera chargé par la première requête avec la bibliothèque SESAM OML PLAM. Les appels suivants réutiliseront le pilote chargé, son catalogue catalog , son schéma schema et son utilisateur user .

Lors de la création d'une base de données, le nom catalog est spécifié dans les directives de configuration SESAM avec la commande //ADD-SQL-DATABASE-CATALOG-LIST ENTRY-1 = *CATALOG(CATALOG-NAME = catalogname,...)

schema référence le schéma de base voulu (voir dans le manuel SESAM).

user spécifie l'un des utilisateurs qui est autorisé à accéder à la combinaison catalog et/ou schema . Notez que user est complètement indépendant de l'utilisateur système et des protections HTTP par mot de passe. Il n'apparaît que dans la configuration SESAM.

Connexion à une base SESAM

```

<?php
if (!sesam_connect("moncatalogue", "monschema", "toto")) {
    die("Impossible de se connecter à SESAM");
}
?>

```

Voir aussi [sesam_disconnect](#) .

8.116.15 **sesam_diagnostic()** : Retourne l'état de la dernière requête SESAM

array **sesam_diagnostic** (void)

sesam_diagnostic retourne un tableau associatif avec l'état et les codes de la dernière requête SQL. Les éléments du tableau sont :

Élément	Contenu
\$array["sqlstate"]	code d'erreur à 5 chiffres (voir le manuel SESAM pour obtenir une description des valeurs possibles de SQLSTATE)

<code>\$array["rowcount"]</code>	nombre de lignes affectées dans la dernière requête immédiate (update/insert/delete) : uniquement après une requête immédiate.
<code>\$array["errmsg"]</code>	message d'erreur lisible : uniquement après une erreur
<code>\$array["errcol"]</code>	numéro de colonne de la dernière erreur (indexée à partir de 0, -1 si indéfinies : uniquement après une erreur).
<code>\$array["errlin"]</code>	numéro de ligne de la dernière erreur (indexée à partir de 0, -1 si indéfinies : uniquement après une erreur).

Dans l'exemple suivant, une erreur de syntaxe (E SEW42AE ILLEGAL CHARACTER) est affichée avec la requête SQL, et en désignant la position de l'erreur :

Afficher une erreur SESAM

```
<?php
// Fonction qui affiche un message d'erreur formaté
// en affichant la position de l'erreur dans le message d'erreur
function PrintReturncode($exec_str)
{
    $serr = Sesam_Diagnostic();
    $colspan=4; // 4 colonnes pour : sqlstate, errlin, errcol, rowcount
    if ($serr["errlin"] == -1) {
        --$colspan;
    }
    if ($serr["errcol"] == -1) {
        --$colspan;
    }
    if ($serr["rowcount"] == 0) {
        --$colspan;
    }
    echo "<table border=\\"1\">\n";
    echo "<tr><th colspan=\\" . $colspan . "\"><span class=\\"spanred\">ERROR:</span> ".
        htmlspecialchars($serr["errmsg"])."</th></tr>\n";
    if ($serr["errcol"] >= 0) {
        echo "<tr><td colspan=\\" . $colspan . "\"><pre>\n";
        $serrstmt = $exec_str.\n";
        for ($lin=0; $serrstmt != ""; ++$lin) {
            if ($lin != $serr["errlin"]) { // $lin est moins ou plus grand qu'errlin
                if (! ($i = strchr($serrstmt, "\n"))) {
                    $i = "";
                }
                $line = substr($serrstmt, 0, strlen($serrstmt)-strlen($i)+1);
                $serrstmt = substr($i, 1);
                if ($line != "\n") {
                    print htmlspecialchars($line);
                }
            } else {
                if (! ($i = strchr($serrstmt, "\n"))) {
                    $i = "";
                }
                $line = substr($serrstmt, 0, strlen($serrstmt)-strlen($i)+1);
                $serrstmt = substr($i, 1);
                for ($col=0; $col < $serr["errcol"]; ++$col) {
                    echo (substr($line, $col, 1) == "\t") ? "\t" : ".";
                }
                echo "<span class=\\"spanred\">\\</span>\n";
                print "<span class=\\"normal\">".htmlspecialchars($line)."</span>";
                for ($col=0; $col < $serr["errcol"]; ++$col) {
                    echo (substr($line, $col, 1) == "\t") ? "\t" : ".";
                }
                echo "<span class=\\"spanred\">/</span>\n";
            }
        }
        echo "</pre></td></tr>\n";
    }
    echo "<tr>\n";
}
```

```

echo " <td>sqlstate=" . $err["sqlstate"] . "</td>\n";
if ($err["errlin"] != -1) {
    echo " <td>errlin=" . $err["errlin"] . "</td>\n";
}
if ($err["errcol"] != -1) {
    echo " <td>errcol=" . $err["errcol"] . "</td>\n";
}
if ($err["rowcount"] != 0) {
    echo " <td>rowcount=" . $err["rowcount"] . "</td>\n";
}
echo "</tr>\n";
echo "</table>\n";
}
if (!sesam_connect("moncatalogue", "phoneno", "toto")) {
    die("cannot connect");
}
$stmt = "SELECT * FROM phone\n".
        " WHERE@ LASTNAME='KRAEMER'\n".
        " ORDER BY FIRSTNAME";
if (! ($result = sesam_query($stmt))) {
    PrintReturncode($stmt);
}
?>

```

Voir aussi [sesam_errormsg](#) pour un accès simplifié aux messages d'erreur.

8.116.16 [sesam_disconnect\(\)](#) : Déconnexion d'une base SESAM

bool [sesam_disconnect](#) (void)

[sesam_disconnect](#) retourne toujours TRUE .

[sesam_disconnect](#) ferme le lien logique à la base de données SESAM (sans réellement déconnecter et démonter le pilote).

Notez que ceci n'est généralement pas nécessaire, car la connexion ouverte est automatiquement fermée à la fin du script. Les données qui ne seront pas validées seront alors annulées, grâce à un [sesam_rollback](#) implicite.

[sesam_disconnect](#) ne ferme pas les connexions persistantes : elle invalide simplement les catalogues catalog , schéma schema et utilisateur user courants, de manière à ce que les prochains appels à des fonctions SESAM échouent.

Exemple avec [sesam_disconnect](#)

```

<?php
if (sesam_connect("moncatalogue", "monschema", "toto")) {
    //... quelques requêtes et d'autres traitements ...
    sesam_disconnect();
}
?>

```

Voir aussi [sesam_connect](#) .

8.116.17 [sesam_errormsg\(\)](#) : Retourne le message d'erreur SESAM

string [sesam_errormsg](#) (void)

sesam_errormsg retourne le message d'erreur SESAM associé à la dernière requête SQL.

Exemple avec sesam_errormsg

```
<?php
if (!sesam_execimm($stmt)) {
    echo sesam_errormsg() . "<br />\n";
}
?>
```

Voir aussi sesam_diagnostic pour la liste complète des états de requêtes SQL.

8.116.18 sesam_execimm() : Exécute immédiatement une requête SQL

string sesam_execimm (string query)

sesam_execimm retourne un identifiant de résultat SESAM en cas de succès, et FALSE sinon.

sesam_execimm exécute immédiatement la requête query (i.e., une requête de type UPDATE, INSERT ou DELETE qui ne retourne aucun résultat, et n'a aucune variable d'entrées ou de sorties). Les requêtes de types "SELECT" ne peuvent pas être utilisées avec la fonction sesam_execimm . sesam_execimm modifie la valeur affected_rows , pour une lecture ultérieure avec sesam_affected_rows .

Notez que sesam_query peut gérer les requêtes immédiates et les requêtes de sélection. Utilisez sesam_execimm uniquement si vous connaissez le type de requête auparavant. Une tentative de requête de sélection avec sesam_execimm retournera \$err["sqlstate"] == "42SBW" .

L'identifiant de résultat retourné ne peut pas être utilisé pour lire quoi que ce soit, mais il peut être passé à sesam_affected_rows ; il n'est retourné que pour symétrie avec la fonction sesam_query .

Exemple avec sesam_execimm

```
<?php
$stmt = "INSERT INTO mytable VALUES ('une', 'deux')";
$result = sesam_execimm($stmt);
$error = sesam_diagnostic();
echo "sqlstate = " . $err["sqlstate"] . "\n".
     "Nombre de lignes affectées = " . $err["rowcount"] . " == " .
     sesam_affected_rows($result) . "\n";
?>
```

Voir aussi sesam_query et sesam_affected_rows .

8.116.19 sesam_fetch_array() : Lit une ligne dans un tableau associatif

array sesam_fetch_array (string result_id , int whence , int offset)

sesam_fetch_array retourne un tableau qui correspond à la ligne lue dans le résultat result_id , ou FALSE s'il n'y a pas d'autres lignes.

sesam_fetch_array est une version alternative de sesam_fetch_row . Au lieu de stocker les données dans un tableau à indice numérique, il enregistre les données dans un tableau associatif, en utilisant les noms des champs comme clés.

`result_id` est un identifiant de résultat valide retourné par `sesam_query` (avec une requête de sélection seulement!).

Pour connaître les valeurs valides des options `whence` et `offset`, reportez-vous à `sesam_fetch_row`.

`sesam_fetch_array` lit une ligne de données dans le résultat `result_id`. La ligne est retournée sous forme d'un tableau associatif. Chaque colonne est enregistrée avec son nom comme index. Les noms des colonnes sont convertis en minuscules.

Les colonnes sans nom (par exemple, les résultats d'opérations arithmétiques) et les champs vides ne sont pas stockés dans ce tableau. De plus, si deux colonnes ont le même nom, la dernière colonne écrasera la précédente. Dans cette situation, utilisez de préférence `sesam_fetch_row` ou bien, faites un alias de la colonne.

Alias de colonnes avec SESAM

```
SELECT TBL1.COL AS FOO, TBL2.COL AS BAR FROM TBL1, TBL2
```

Une gestion spéciale permet de lire les champs multiples, qui sinon, auraient toutes le même nom. Pour chaque colonne d'un champ multiple, le nom d'index est créé en ajoutant le numéro de sous-index à la suite du nom de la colonne. Ces sous-index sont numérotés à partir de 1.

Lire les champs multiples de SESAM

```
CREATE TABLE ... ( ... MULTI(3) INT )
```

Les index associatifs utilisés pour les valeurs individuelles du champ multiple sont : "multi(1)", "multi(2)", et "multi(3)", respectivement.

Les prochains appels à `sesam_fetch_array` liront la prochaine ligne (ou la précédente, ou la n-ième, suivant les attributs de défilement), jusqu'à ce qu'il n'y ait plus de lignes.

Exemple avec `sesam_fetch_array`

```
<?php
$result = sesam_query("SELECT * FROM phone\n" .
                    " WHERE LASTNAME='" . strtoupper($name) . "'\n" .
                    " ORDER BY FIRSTNAME", 1);

if (!$result) {
    // erreur
}
// Affiche la table
echo "<table border=\\"1\">\n";
while (($row = sesam_fetch_array($result)) && count($row) > 0) {
    echo "<tr>\n";
    echo "<td>" . htmlspecialchars($row["firstname"]) . "</td>\n";
    echo "<td>" . htmlspecialchars($row["lastname"]) . "</td>\n";
    echo "<td>" . htmlspecialchars($row["phoneno"]) . "</td>\n";
    echo "</tr>\n";
}
echo "</table>\n";
sesam_free_result($result);
?>
```

Voir aussi `sesam_fetch_row` qui retourne un tableau numérique.

8.116.20 `sesam_fetch_result()` : Retourne tout ou partie d'un résultat SESAM

mixed `sesam_fetch_result` (string `result_id` , int `max_rows`)

8.116.19 `sesam_fetch_array()` : Lit une ligne dans un tableau associatif

sesam_fetch_result retourne un tableau avec les lignes du résultat identifié par `result_id` , éventuellement limité à un maximum de `max_rows` Notez que les lignes et les colonnes sont indexées à partir de 0.

Elément du tableau	Contents
<code>int \$arr["count"]</code>	Nombre de colonnes dans le résultat (ou zéro si c'était une requête immédiate).
<code>int \$arr["rows"]</code>	Nombre de lignes dans le résultat (entre zéro et <code>max_rows</code>)
<code>bool \$arr["truncated"]</code>	TRUE si le nombre de lignes était d'au moins <code>max_rows</code> , FALSE sinon. Notez que même si cette valeur est à TRUE , le prochain appel à <u>sesam_fetch_result</u> peut ne retourner aucune ligne parce qu'il n'y a plus d'entrées.
<code>mixed \$arr[col][row]</code>	les valeurs du résultat à la ligne <code>row</code> et colonne <code>col</code> . Le résultat est un tableau multidimensionnel. <code>row</code> va de 0 à <code>\$arr["rows"]-1</code> , et <code>col</code> de 0 à <code>\$arr["count"]-1</code>). Les champs peuvent être vides : vous devez vérifier leur existence avec la fonction <u>isset</u> . Le type retourné dépend du type SQL déclaré pour cette colonne (voir <u>Introduction SESAM</u> pour connaître les conversions utilisées). Les champs multiples SESAM sont traités comme des séquences de colonnes.

Notez que la quantité de mémoire utilisée par des requêtes peut se révéler gigantesque. Utilisez alors `max_rows` pour limiter le nombre maximum de lignes retournées, à moins que vous ne soyez absolument sûr que votre résultat ne consommera pas toute la mémoire disponible.

Voir aussi sesam_fetch_row et sesam_field_array pour vérifier les champs multiples. Voyez sesam_query pour un exemple complet avec sesam_fetch_result .

8.116.21 sesam_fetch_row() : Lit une ligne dans un tableau

`array sesam_fetch_row (string result_id , int whence , int offset)`

sesam_fetch_row retourne un tableau qui correspond à la ligne lue dans le résultat `result_id` , ou FALSE s'il n'y a plus de ligne.

Le nombre de colonnes du résultat est retourné dans un élément du tableau associatif retourné `$array["count"]`. Comme certaines lignes peuvent être vides, la fonction count ne peut être utilisée avec le tableau ainsi retourné par sesam_fetch_row .

`result_id` est un identifiant de résultat valide retourné par sesam_query (avec une requête de selection seulement !).

`whence` est un paramètre optionnel lors d'une opération de lecture sur un curseur à défilement, qui peut prendre une des valeurs suivantes :

Valeur	Constante	Signification
0	SESAM SEEK NEXT	Lecture séquentielle (après la lecture, la position est déplacée à SESAM SEEK NEXT)
1	SESAM SEEK PRIOR	Lecture séquentielle à rebours (après la lecture, la position est déplacée à SESAM SEEK PRIOR)
2	SESAM SEEK FIRST	Repositionnement au début (après la lecture, la position est déplacée à SESAM SEEK NEXT)
3	SESAM SEEK LAST	Repositionnement à la fin (après la lecture, la position est déplacée à SESAM SEEK PRIOR)
4	SESAM SEEK ABSOLUTE	

		Repositionnement absolu à offset (index commençant à 0. Après la lecture, la position est placée à <code>SESAM_SEEK_ABSOLUTE</code> , et le pointeur interne est auto-incrémenté).
5	<code>SESAM_SEEK_RELATIVE</code>	Repositionnement relatif à offset, où offset peut être positif ou négatif

Ce paramètre n'est valable que pour les curseurs à défilement.

Lors de l'utilisation de curseurs à défilement, le curseur peut être librement repositionné. Si le paramètre `whence` est omis, les valeurs par défaut seront utilisées (initialisées à : `SESAM_SEEK_NEXT`, et modifiées par `sesam_seek_row`). Si `whence` est fourni, sa valeur remplacera les valeurs par défaut.

offset est un paramètre optionnel qui n'est utilisé (et nécessaire) que si `whence` vaut soit `SESAM_SEEK_RELATIVE` ou `SESAM_SEEK_ABSOLUTE`. Ce paramètre n'est valable que pour les curseurs à défilement.

`sesam_fetch_row` lit une ligne de données dans le résultat `result_id`. La ligne est retournée sous forme d'un tableau (indexé de 0 à `$array["count"]-1`). Les champs peuvent être vides : il faut vous assurer de leur existence en utilisant la fonction `isset`. Le type de la valeur retournée dépend du type SQL déclaré dans la base (voir [introduction SESAM](#) pour connaître les conversions utilisées). Les champs multiples SESAM sont linéarisés, et traités comme autant de colonnes.

Les prochains appels à `sesam_fetch_row` liront la prochaine ligne (ou la précédente, ou la n-ième, suivant le type de défilement) dans le résultat, ou `FALSE`, s'il n'y a plus de lignes.

Exemple avec `sesam_fetch_row`

```
<?php
$result = sesam_query ("SELECT * FROM phone\n".
    " WHERE LASTNAME='".strtoupper($name)."' \n".
    " ORDER BY FIRSTNAME", 1);

if (!$result) {
    // erreur
}

// Affiche la table dans l'ordre inverse
echo "<table border='1'\n";
$row = sesam_fetch_row ($result, SESAM_SEEK_LAST);
while (is_array($row)) {
    echo "<tr>\n";
    for($col = 0; $col < $row["count"]; ++$col) {
        echo "<td>" . htmlspecialchars($row[$col]) . "</td>\n";
    }
    echo "</tr>\n";
    // utilise la valeur implicite de SESAM_SEEK_PRIOR
    $row = sesam_fetch_row ($result);
}
echo "</table>\n";
sesam_free_result ($result);
?>
```

Voir aussi `sesam_fetch_array` qui retourne un tableau associatif et `sesam_fetch_result` qui retourne plusieurs lignes en même temps.

8.116.22 `sesam_field_array()` : Retourne des informations sur une colonne de résultat

array **sesam_field_array** (string result_id)

result_id est un identifiant de résultat valide retourné par la fonction [sesam_query](#) .

[sesam_field_array](#) retourne un tableau contenant les informations (nom de colonne, type, précision...) sur une colonne dans le résultat associé à result_id .

Index	Contenu
int \$arr["count"]	Nombre total de colonnes dans le résultat (ou zéro si la requête était immédiate). Les champs multiples de SESAM sont linéarisés, et traités comme autant de colonnes.
string \$arr[col]["name"]	Le nom de la colonne col , avec col qui vaut entre 0 et \$arr["count"]-1 . La valeur retournée peut être une chaîne vide (pour les colonnes dynamiquement générées). Les champs multiples SESAM sont linéarisés, et traités comme autant de colonnes, avec le même nom.
string \$arr[col]["count"]	L'attribut "count" décrit le facteur de répétition quand la colonne a été déclarée comme un champ multiple. Généralement, cet attribut est à 1. La première colonne d'un champ multiple contient le nombre de répétitions, tandis que les colonnes suivantes ont un facteur de répétition mis à 1. Ceci peut être utilisé pour détecter les champs multiples. Reportez-vous à l'exemple de la fonction sesam_query pour avoir un exemple d'utilisation.
string \$arr[col]["type"]	Type de variable PHP pour les données de la colonne col , où col vaut de 0 à \$arr["count"]-1 . La valeur retournée peut être l'une de celles-ci : <ul style="list-style-type: none"> • "integer" • "double" • "string" , suivant le type de données SQL. Les champs multiples SESAM sont linéarisés et traités comme autant de colonnes ayant le même type PHP.
string \$arr[col]["sqltype"]	Type de données SQL de la colonne col , où col vaut de 0 à \$arr["count"]-1 . La valeur retournée peut être l'une de celles-ci : <ul style="list-style-type: none"> • "CHARACTER" • "VARCHAR" • "NUMERIC" • "DECIMAL" • "INTEGER" • "SMALLINT" • "FLOAT" • "REAL" • "DOUBLE" • "DATE" • "TIME" • "TIMESTAMP" , décrivant le type de données SQL. Les champs multiples SESAM sont linéarisés et traités comme autant de colonnes du même type.
string \$arr[col]["length"]	La taille de l'attribut, au sens SQL, de la colonne col , où col vaut de 0 à \$arr["count"]-1 . La longueur est utilisée avec les champs "CHARACTER" et "VARCHAR", pour spécifier la taille maximale de la colonne. Les champs multiples SESAM sont linéarisés et traités comme autant de colonnes ayant la même taille SQL.

string \$arr[col]["precision"]	La précision de la colonne col , au sens SQL, où col vaut de 0 à \$arr["count"]-1 . La précision est utilisée avec les champs numériques et de date. Les champs multiples SESAM sont linéarisés et traités comme autant de colonnes ayant la même précision SQL.
string \$arr[col]["scale"]	L'échelle de la colonne col , au sens SQL, où col vaut de 0 à \$arr["count"]-1 . L'échelle est utilisée avec les champs numériques. Les champs multiples SESAM sont linéarisés et traités comme autant de colonnes ayant la même échelle SQL.

Voir aussi [sesam_query](#) , pour un exemple d'utilisation de [sesam_field_array](#) .

8.116.23 [sesam_field_name\(\)](#) : Retourne le nom d'une colonne

int [sesam_field_name](#) (string result_id , int index)

[sesam_field_name](#) retourne le nom du champ index dans le résultat identifié par result_id , ou FALSE en cas d'erreur.

Pour les requêtes immédiates, ou les colonnes dynamiques, une chaîne vide est retournée.

Note
Les colonnes sont indexées à partir de 0, et non pas 1.

Voir aussi : [sesam_field_array](#) . Cette fonction fournit une interface simple aux noms et types de colonnes, et permet la détection des champs multiples.

8.116.24 [sesam_free_result\(\)](#) : Libère les ressources SESAM

int [sesam_free_result](#) (string result_id)

[sesam_free_result](#) libère les ressources réservées par la requête result_id . Retourne FALSE en cas d'erreur.

8.116.25 [sesam_num_fields\(\)](#) : Retourne le nombre de colonnes dans un résultat

int [sesam_num_fields](#) (string result_id)

Après avoir appelé [sesam_query](#) avec une requête de sélection, [sesam_num_fields](#) indique le nombre de colonnes du résultat identifié par result_id . Retourne FALSE en cas d'erreur.

Pour les requêtes immédiates, la valeur zéro est retournée. Les champs multiples SESAM comptent autant que leurs tailles respectives, c'est-à-dire qu'un champ multiple de trois colonnes comptera comme trois colonnes.

Voir aussi [sesam_query](#) et [sesam_field_array](#) pour savoir distinguer les champs multiples des colonnes standard.

8.116.26 `sesam_query()` : Exécute une requête SESAM

string `sesam_query` (string `query` , *bool* `scrollable`)

`sesam_query` retourne un identifiant de résultat SESAM en cas de succès, ou FALSE en cas d'erreur.

L'identifiant de résultat est utilisé par d'autres fonctions SESAM pour lire les valeurs.

`sesam_query` envoie une requête à la base active. Elle peut exécuter aussi bien une requête immédiate (DELETE, UPDATE ou INSERT), ou une requête de sélection. Si une requête immédiate est exécutée, aucun curseur n'est alloué, et il ne sera pas possible d'utiliser les fonctions `sesam_fetch_row` ou `sesam_fetch_result`. Pour les requêtes de sélection, un descripteur de résultat et un curseur (scrollable ou séquentiel, suivant le paramètre optionnel `scrollable` passé) sera alloué. Si `scrollable` est omis, le curseur sera séquentiel.

Lorsque vous utilisez les curseurs à défilement, le curseur peut être positionné librement dans le résultat. Pour chaque requête à défilement, il existe des valeurs globales de types de défilement (initialisée à : `SESAM_SEEK_NEXT`) et la position peut être fixée une seule fois par `sesam_seek_row` ou bien à chaque appel, avec la fonction `sesam_fetch_row`.

Pour les requêtes immédiates, le nombre de lignes affectées est sauvé, et est accessible par la fonction `sesam_affected_rows`.

Liste toutes les lignes de table "phone" sous forme de table HTML

```
<?php
if (!sesam_connect("phonedb", "demo", "toto")) {
    die("cannot connect");
}
$result = sesam_query("select * from phone");
if (!$result) {
    $err = sesam_diagnostic();
    die ($err["errmsg"]);
}
echo "<table border=\"1\">\n";
// Ajoute l'en-tête de titre comme nom de colonne
if ($cols = sesam_field_array($result)) {
    echo "<tr><th colspan=\"".$cols["count"]."\">Résultat:</th></tr>\n";
    echo "<tr>\n";
    for ($col = 0; $col < $cols["count"]; ++$col) {
        $colattr = $cols[$col];
        /* étend les en-têtes de la table au-dessus des champs multiples */
        if ($colattr["count"] > 1) {
            echo "<th colspan=\"".$colattr["count"]."\">".$colattr["name"].
                "(1..".$colattr["count"].")</th>\n";
            $col += $colattr["count"] - 1;
        } else {
            echo "<th>". $colattr["name"] . "</th>\n";
        }
    }
    echo "</tr>\n";
}

do {
    // Lit les résultats par bloc de 100
    $ok = sesam_fetch_result($result,100);
    for ($row=0; $row < $ok["rows"]; ++$row) {
        echo "<tr>\n";
        for ($col = 0; $col < $ok["cols"]; ++$col) {
            if (isset($ok[$col][$row])) {
```

```

        echo "<td>" . $ok[$col][$row] . "</td>\n";
    } else {
        echo "<td>-empty-</td>\n";
    }
}
echo "</tr>\n";
}
} while ($ok["truncated"]); // tant qu'il a y encore des données

echo "</table>\n";
// libère les ressources
sesam_free_result($result);
?>

```

Voir aussi : [sesam_fetch_row](#) et [sesam_fetch_result](#) .

8.116.27 [sesam_rollback\(\)](#) : Annule une transaction SESAM

bool [sesam_rollback](#) (void)

[sesam_rollback](#) retourne TRUE en cas de succès et FALSE en cas d'erreur.

[sesam_rollback](#) annule toutes les modifications en cours sur la base. Les curseurs de résultats et les descripteurs de résultats seront affectés.

A la fin de chaque script, et dans chaque appel à [sesam_disconnect](#) , un appel implicite à [sesam_rollback](#) est fait, annulant toutes les transactions non validées dans la base.

Annulation d'une transaction SESAM

```

<?php
if (sesam_connect ("moncatalogue", "monschema", "toto")) {
    if (sesam_execimm("INSERT INTO matable VALUES (*, 'Petit Test', <0, 8, 15>")
        && sesam_execimm("INSERT INTO autretable VALUES (*, 'Autre Test', 1)")) {
        sesam_commit();
    } else {
        sesam_rollback();
    }
}
}
?>

```

Voir aussi [sesam_commit](#) .

8.116.28 [sesam_seek_row\(\)](#) : Déplace un curseur à défilement

bool [sesam_seek_row](#) (string result_id , int whence , int offset)

result_id est un indentifiant de résultat valide (requête de sélection, et curseur à défilement créé avec [sesam_query](#)).

whence modifie la valeur globale par défaut pour le type de défilement, spécifie le type de défilement à utiliser lors des opérations de lectures ultérieurs. Les valeurs valides sont les suivantes :

Valeur	Constante	Signification
0	SESAM_SEEK_NEXT	Lecture séquentielle (après la lecture, la position est déplacée

		à SESAM_SEEK_NEXT)
1	SESAM_SEEK_PRIOR	Lecture séquentielle à rebours (après la lecture, la position est déplacée à SESAM_SEEK_PRIOR)
2	SESAM_SEEK_FIRST	Repositionnement au début (après la lecture, la position est déplacée à SESAM_SEEK_NEXT)
3	SESAM_SEEK_LAST	Repositionnement à la fin (après la lecture, la position est déplacée à SESAM_SEEK_PRIOR)
4	SESAM_SEEK_ABSOLUTE	Repositionnement absolu à offset (index commençant à 0. Après la lecture, la position est placée à SESAM_SEEK_ABSOLUTE , et le pointeur interne est auto-incrémenté).
5	SESAM_SEEK_RELATIVE	Repositionnement relatif à offset , où offset peut être positif ou négatif

offset est optionnel. Il ne sert que lorsque whence vaut soit SESAM_SEEK_RELATIVE , soit SESAM_SEEK_ABSOLUTE .

8.116.29 **sesam_settransaction()** : Modifie les paramètres de transaction SESAM

bool **sesam_settransaction** (int isolation_level , int read_only)

sesam_settransaction retourne TRUE si les valeurs sont valides et que la modification a été réussie. FALSE sinon.

sesam_settransaction remplace les valeurs par défaut du niveau d'isolation ("isolation level") et de lecture seule ("read-only") fixées par le fichier de configuration SESAM), afin d'optimiser les requêtes ultérieures et garantir la cohérence de la base. Ces valeurs ne seront utilisées que pour la prochaine transaction.

sesam_settransaction ne peut être appelée qu'avant le début de la transaction. Elle est inefficace si la transaction a déjà commencé.

Pour simplifier l'utilisation de cette fonction dans les scripts PHP, les constantes suivantes ont été définies en PHP (reportez-vous au manuel SESAM pour avoir des détails sur leur signification) :

Valeur	Constante	Signification
1	SESAM_TXISOL_READ_UNCOMMITTED	Lecture sans validation
2	SESAM_TXISOL_READ_COMMITTED	Lecture avec validation
3	SESAM_TXISOL_REPEATABLE_READ	Lecture récurrente
4	SESAM_TXISOL_SERIALIZABLE	Sérialisable
Valeur	Constante	Signification
0	SESAM_TXREAD_READWRITE	Lecture/écriture
1	SESAM_TXREAD_READONLY	Lecture seule

Les valeurs modifiées par sesam_settransaction remplaceront les valeurs par défaut spécifiées dans le fichier de configuration SESAM.

<p>Modifier les paramètres de configuration SESAM</p> <pre><?php sesam_settransaction(SESAM_TXISOL_REPEATABLE_READ, SESAM_TXREAD_READONLY); ?></pre>
--

8.117 Sessions

8.117.1 Introduction

Le support des sessions de PHP est un moyen de préserver des données entre plusieurs accès. Cela vous permet de créer des applications personnalisées, et d'augmenter l'attrait de votre site.

Chaque visiteur accédant à votre page web se voit assigner un identifiant unique, appelé "identifiant de session". Il peut être stocké soit dans un cookie, soit propagé dans l'URL.

Le support des sessions vous permet d'enregistrer un nombre illimité de variables qui doivent être préservées entre les requêtes. Lorsqu'un visiteur accède à votre site, PHP va vérifier automatiquement (si `session.auto_start` est activé) ou sur demande (explicitement avec `session_start` ou implicitement avec `session_register`) s'il existe une session du même nom. Si c'est le cas, l'environnement précédemment sauvé sera recréé.

Attention

Si vous activez <code>session.auto_start</code> , alors vous ne pourrez pas enregistrer d'objets dans votre session tant que la définition de la classe ne sera pas chargée avant le début de la session, pour recréer les objets de votre session.

Toutes les variables sont sérialisées après l'exécution du script PHP. Les variables qui sont indéfinies sont marquées comme telles. Lors des accès ultérieurs, elles ne seront pas définies, jusqu'à ce que l'utilisateur le fasse.

Attention

Quelques types de données ne peuvent pas être linéarisés pour être stockés dans les sessions. Cela inclut les variables de type ressource ou les objets avec des références circulaires (i.e. objet qui passe une référence à lui-même à un autre objet).

Note

La gestion des sessions a été ajoutée en PHP 4.0.

Note

Notez que lorsque vous travaillez avec les sessions, un enregistrement dans la session ne sera pas créé tant que la variable ne sera pas enregistrée en utilisant la fonction <code>session_register</code> ou en ajoutant une clé à la variable super-globale <code>\$_SESSION</code> . Cela n'est vrai que si vous avez débuté une session en appelant la fonction <code>session_start</code> .

8.117.2 Sessions et sécurité

Lien externe : [Session fixation](#)

Utiliser les sessions ne signifie pas que les données de session ne pourront être vues que par un seul utilisateur. Il est important de garder cela en tête lorsque vous stockez et affichez des données importantes. Lorsque vous stockez des données dans une session, il faut se demander quels seront les problèmes posés si quelqu'un d'autre accède à cette information, ou comment votre application est affectée si la session est en fait celle d'un autre.

Par exemple, si quelqu'un usurpe une session, il peut alors poster un message dans un forum sous une fausse identité. Quelle est la gravité de ce problème ? Ou bien, il peut accéder aux commandes d'un client, et même, modifier son panier d'achat. A priori, c'est moins problématique pour un fleuriste que pour un pharmacien. Si vous voulez résoudre ce souci de façon simple, il peut être utile d'activer `session.use_only_cookies`. Dans ce cas, les cookies devront être activés par le client,

sinon, les sessions ne fonctionneront pas.

Par conséquent, lorsque vous manipulez des données importantes, il faut exploiter d'autres méthodes pour décider si une session est valide ou pas. Les sessions ne fournissent pas une méthode fiable d'identification.

Les sessions reposent sur un identifiant de session, ce qui signifie que quelqu'un peut voler cet identifiant, rien qu'en volant l'ID. Ce vol peut être rendu très difficile, comme en utilisant les cookies, mais en aucun cas cela sera impossible. Les sessions dépendent aussi de la discipline de l'utilisateur qui referme son navigateur à la fin de la session pour tout clore proprement. De plus, même les cookies de session peuvent être surveillés sur un réseau, ou bien notés par un proxy car ils transitent en clair sur le réseau. Pour remédier à cela, vous devriez implémenter un chiffrement SSL sur votre plate-forme.

8.117.3 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

Note
Optionnellement, vous pouvez utiliser l'allocation de mémoire partagée (mm), développé par Ralf S.Engelschall, pour stocker votre session. Vous devez télécharger mm et l'installer. Cette option n'est pas disponible pour les environnements Windows. Notez que le module de stockage de session mm ne garantit pas les verrous de sessions en cas d'accès multiples à la même session. Il peut être plus approprié d'utiliser un système de fichiers basé en mémoire partagée (comme tmpfs sur Solaris/Linux ou /dev/md sur BSD) pour stocker les sessions dans des fichiers, car ils seront proprement verrouillés.

8.117.4 Installation

Le support des sessions est activé par défaut. Si vous souhaitez l'exclure de PHP, vous devez utiliser l'option `--disable-session` lors de l'exécution du script de configuration. Pour utiliser la mémoire vive pour le stockage des sessions, compilez PHP avec l'option `--with-mm[=DIR]`.

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de bibliothèque supplémentaire pour disposer de ces fonctions.

Note
Par défaut, toutes les données relatives à une session particulière seront stockées dans un fichier du répertoire spécifié par <code>session.save_path</code> dans les options du fichier <code>php.ini</code> . Un fichier pour chaque session sera créé. Cela est dû au fait que une session est ouverte (un fichier est créé) mais aucune donnée n'est écrite dans ce fichier. Notez que ce comportement est un effet des limitations d'utilisation du système de fichiers et il est possible qu'un gestionnaire de session personnalisé (par exemple, un qui utilise une base de données) ne garde aucune trace des sessions où aucune donnée n'y a été enregistrée.

8.117.5 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable	Historique
<code>session.save_path</code>	<code>"/tmp"</code>	<code>PHP_INI_ALL</code>	

session.name	"PHPSESSID"	PHP_INI_ALL	
session.save_handler	"files"	PHP_INI_ALL	
session.auto_start	"0"	PHP_INI_ALL	
session.gc_probability	"1"	PHP_INI_ALL	
session.gc_divisor	"100"	PHP_INI_ALL	Disponible de PHP 4.3.2.
session.gc_maxlifetime	"1440"	PHP_INI_ALL	
session.serialize_handler	"php"	PHP_INI_ALL	
session.cookie_lifetime	"0"	PHP_INI_ALL	
session.cookie_path	"/"	PHP_INI_ALL	
session.cookie_domain	""	PHP_INI_ALL	
session.cookie_secure	""	PHP_INI_ALL	Disponible de PHP 4.0.4.
session.use_cookies	"1"	PHP_INI_ALL	
session.use_only_cookies	"0"	PHP_INI_ALL	Disponible de PHP 4.3.0.
session.referer_check	""	PHP_INI_ALL	
session.entropy_file	""	PHP_INI_ALL	
session.entropy_length	"0"	PHP_INI_ALL	
session.cache_limiter	"nocache"	PHP_INI_ALL	
session.cache_expire	"180"	PHP_INI_ALL	
session.use_trans_sid	"0"	PHP_INI_ALL	PHP_INI_ALL PHP <= 4.2.3 PHP_INI_PER en PHP < 5. Disponible de PHP 4.0.3.
session.bug_compat_42	"1"	PHP_INI_ALL	Disponible de PHP 4.3.0.
session.bug_compat_warn	"1"	PHP_INI_ALL	Disponible de PHP 4.3.0.
session.hash_function	"0"	PHP_INI_ALL	Disponible de PHP 5.0.0.
session.hash_bits_per_character	"4"	PHP_INI_ALL	Disponible de PHP 5.0.0.
url_rewriter.tags	"a=href,area=href,frame=src,form=,fieldset="	PHP_INI_ALL	Disponible de PHP 4.0.4.

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [Directives du php.ini](#) .

Le système de sessions dispose d'un grand nombre de directives dans le fichier php.ini . En voici une présentation :

session.save_handler string

Définit le nom du gestionnaire de session qui est utilisé pour stocker et relire les données. Par défaut, c'est le système intégré par fichiers : files . Voir aussi [session_set_save_handler](#) .

session.save_path string

Définit le chemin qui doit être passé au gestionnaire de sauvegarde. Si vous décidez de choisir le gestionnaire par défaut (par fichiers), cet argument sera utilisé comme

dossier de sauvegarde des sessions. Par défaut, il vaut /tmp . Voir aussi [session_save_path](#) .

Il y a un argument optionnel N à cette directive qui détermine la profondeur de répertoires où votre fichier de session sera stocké. Par exemple, si vous définissez '5;/tmp' , votre fichier sera situé dans /tmp/4/b/1/e/3/sess_4b1e384ad74619bd212e236e52a5a174lf . Si vous voulez utiliser N , vous devez créer tous ces répertoires avant de les utiliser. Un petit script shell existe dans ext/session pour réaliser ces créations et il se nomme mod_files.sh . Notez également que si N est utilisé et est supérieur à 0, alors la routine automatique gc (garbage collection) ne sera pas exécutée ; voir une copie de php.ini pour plus d'informations. Egalement, si vous utilisez N , assurez-vous d'entourer session.save_path de "doubles guillemets" car le séparateur (;) est également utilisé pour les commentaires dans php.ini .

Attention
Si vous laissez cette option configurée avec un dossier accessible en lecture à tout le monde, comme /tmp (par défaut), les autres utilisateurs pourront exploiter ces sessions en obtenant la liste de fichiers dans ce dossier.
Note
Avant PHP 4.3.6;, les utilisateurs de Windows doivent changer cette valeur de variable pour que les fonctions de sessions de PHP fonctionnent. Indiquez un chemin de dossier valide, par exemple : c:/temp .

session.name string

Spécifie le nom de la session, qui sera utilisé comme nom de cookie. Il ne doit contenir que des caractères alphanumériques. Par défaut, c'est PHPSESSID . Voir aussi [session_name](#) .

session.auto_start boolean

Spécifie si le module de session doit démarrer automatiquement au début de chaque script PHP. Par défaut, c'est 0 (désactivé).

session.serialize_handler string

Définit le nom du gestionnaire qui est utilisé pour linéariser/délinéariser les données. Actuellement, un format interne à PHP (nommé php) et WDDX (nommé wddx) sont supportés. WDDX est seulement disponible, si PHP a été compilé avec l'option [WDDX](#) . Par défaut, c'est php .

session.gc_probability entier

Spécifie la probabilité, exprimée en pourcentage, en conjonction de session.gc_divisor , que la routine gc (garbage collection) soit démarrée à chaque requête. La valeur par défaut est 1 . Voir [session.gc_divisor](#) pour plus de détails.

session.gc_divisor entier

session.gc_divisor en conjonction avec session.gc_probability définit la probabilité que la routine gc (garbage collection) soit démarrée à chaque début de session. La probabilité est calculée en utilisant gc_probability/gc_divisor, e.g 1/100 signifie qu'il y a 1 % de chance pour que la routine gc démarre à chaque requête. La valeur par défaut est 100 .

session.gc_maxlifetime entier

Spécifie la durée de vie des données sur le serveur, en nombre de secondes. Après cette durée, les données seront considérées comme obsolètes, et supprimées.

Note
Si des scripts différents ont des valeurs différentes de session.gc_maxlifetime mais partagent le même endroit pour y stocker les données de session, alors, le script dont la valeur est la plus petite effacera la donnée. Dans ce cas, utilisez cette directive conjointement avec session.save_path .

Note

Si vous utilisez le gestionnaire de sessions par fichier, qui est fourni par défaut votre système doit garder la trace des dates de dernier accès aux fichiers (atime). La FAT de Windows ne le fait pas, il vous faudra donc trouver un autre système pour gérer les sessions qui ont expiré. Depuis PHP 4.2.3, on utilise mtime (date de modification) au lieu de atime. Ainsi, vous n'aurez plus de souci avec les systèmes de fichiers qui ne gèrent pas atime.

`session.referer_check` entier

Contient une sous-chaîne que vous souhaitez retrouver dans tous les en-têtes HTTP Referer. Si cet en-tête a été envoyé par le client, et que la sous-chaîne n'a pas été trouvée, l'identifiant de session sera considéré comme invalide. Par défaut, cette option est une chaîne vide.

`session.entropy_file` string

Est un chemin jusqu'à une source externe (un fichier), qui sera utilisée comme source additionnelle d'entropie pour la création de l'identifiant de session. Des exemples valides sont `/dev/random` et `/dev/urandom`, qui sont disponibles sur tous les systèmes Unix.

`session.entropy_length` entier

Spécifie le nombre d'octets qui seront lus dans le fichier défini ci-dessus. Par défaut, il vaut 0, c'est à dire inactif.

`session.use_cookies` boolean

Spécifie si le module utilisera les cookies pour stocker les données de session sur le client. Par défaut, il vaut 1, c'est à dire actif.

`session.use_only_cookies` boolean

Spécifie si le module doit utiliser **seulement** les cookies pour stocker les identifiants de sessions du côté du navigateur. Par défaut, cette option vaut 0 (inactif, pour une compatibilité ascendante). En l'activant, vous éviterez les attaques qui utilisent des identifiants de sessions dans les URL. Cette configuration a été ajoutée en PHP 4.3.0.

`session.cookie_lifetime` entier

Spécifie la durée de vie du cookie en secondes. La valeur de 0 signifie : "Jusqu'à ce que le navigateur soit éteint". La valeur par défaut est : 0. Voir aussi [session_get_cookie_params](#) et [session_set_cookie_params](#).

`session.cookie_path` string

Spécifie le chemin utilisé lors de la création du cookie. Par défaut, il vaut `/`. Voir aussi [session_get_cookie_params](#) et [session_set_cookie_params](#).

`session.cookie_domain` string

Spécifie le domaine utilisé lors de la création du cookie. Par défaut, il ne vaut rien, cela signifie que c'est le nom de l'hôte du serveur qui génère le cookie en accord avec les spécifications sur les cookies. Voir aussi [session_get_cookie_params](#) et [session_set_cookie_params](#).

`session.cookie_secure` boolean

Spécifie que les cookies ne doivent être émis que sur des connexions sécurisée. Par défaut, cette option est à off. Cette option a été ajoutée en PHP 4.0.4. Voir aussi [session_get_cookie_params](#) et [session_set_cookie_params](#).

`session.cache_limiter` string

Spécifie le type de contrôle de cache utilisé pour les pages avec sessions. Les valeurs possibles sont : none, nocache, private, private_no_expire, public. Par défaut, il vaut nocache. Voir aussi [session_cache_limiter](#).

`session.cache_expire` entier

Spécifie la durée de vie des données de sessions, en minutes. Cette option n'a aucune conséquence sur le contrôle de cache. Par défaut, il vaut 180 (3 heures). Voir aussi [session_cache_expire](#).

`session.use_trans_sid` boolean

Spécifie si le support du SID est transparent ou pas. Par défaut vaut 0 (désactivé).

Note
En PHP 4.1.2 ou plus ancien, cette option est activée en utilisant l'option de compilation <code>--enable-trans-sid</code> . Depuis PHP 4.2.0, cette option est toujours activée.
Le système de gestion des sessions par URL pose un risque supplémentaire de sécurité : un utilisateur peut envoyer son URL avec l'identifiant de session par email à un ami, ou bien le mettre dans ses signets. Cela diffusera alors l'identifiant de session.

`session.bug_compat_42` boolean

Les versions de PHP antérieures à la version 4.2.3 disposaient d'une fonctionnalité/bogue non documentée, qui vous permettait d'initialiser une variable de session dans le contexte global, même si `register_globals` était désactivé. PHP 4.3.0 et plus récent vous préviendra de l'utilisation de cette fonctionnalité si vous avez aussi activé `session.bug_compat_warn` . Cette fonctionnalité/bogue peut être désactivée en désactivant cette directive.

`session.bug_compat_warn` boolean

Les versions de PHP antérieures à la version 4.2.3 disposaient d'une fonctionnalité/bogue non documentée, qui vous permettait d'initialiser une variable de session dans le contexte global, même si `register_globals` était désactivé. PHP 4.3.0 et plus récent vous préviendra de l'utilisation de cette fonctionnalité si vous avez activé `session.bug_compat_42` et `session.bug_compat_warn` .

`session.hash_function` entier

`session.hash_function` vous permet de spécifier la fonction de hachage à utiliser pour générer les identifiants de session. '0' signifie MD5 (128 bits) et '1' signifie SHA-1 (160 bits).

Note
Cette directive a été ajoutée en PHP 5.

`session.hash_bits_per_character` entier

`session.hash_bits_per_character` vous permet de définir le nombre de bits utilisés pour chaque caractère lors des conversions des données binaires en éléments lisibles. Les valeurs possibles sont '4' (0-9, a-f), '5' (0-9, a-v), et '6' (0-9, a-z, A-Z, "-", ";").

Note
Cette directive a été ajoutée en PHP 5.

`session.url_rewriter.tags` string

Spécifie quelles sont les balises HTML qui doivent être réécrites si le support transparent du SID est activé. Par défaut, il vaut `a=href,area=href,frame=src,input=src,form=fakeentry,fieldset=` .

Note
Si vous voulez vous conformer aux spécifications XHTML, supprimez l'entrée <code>form</code> et utilisez le tag <code><fieldset></code> autour de votre balise <code>form</code> .

Les options `track_vars` et `register_globals` influencent le comportement des sessions, leur stockage et leur restauration.

Note
Depuis PHP 4.0.3, <code>track_vars</code> est toujours activé.

8.117.6 Types de ressources

Cette extension ne définit aucune ressource.

8.117.7 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

SID (chaîne de caractères)

Constante contenant le nom de la session et l'identifiant en cours, sous la forme "name=ID" ou une chaîne vide si l'identifiant de session a été défini dans un cookie de session.

8.117.8 Exemples

Note

Depuis PHP 4.1.0, `$_SESSION` est disponible comme variable globale, au même titre que `$_POST`, `$_GET`, `$_REQUEST`, etc. Contrairement à `$HTTP_SESSION_VARS`, `$_SESSION` est toujours globale. Par conséquent, vous n'avez pas besoin d'utiliser le mot réservé `global` avec `$_SESSION`. Notez que cette documentation a été modifiée pour utiliser `$_SESSION`. Vous pouvez toujours le remplacer par `$HTTP_SESSION_VARS` si vous préférez l'ancienne version. Notez également que vous devez démarrer votre session en utilisant la fonction `session_start` avant d'utiliser la variable super-globale `$_SESSION`.

Les clés du tableau `$_SESSION` sont sujettes aux mêmes limitations que les variables PHP habituelles, c'est-à-dire qu'elles ne peuvent pas commencer par un nombre, mais commencer par une lettre ou un souligné `'_'`. Pour plus de détails, reportez-vous à la section sur les [variables](#).

Si `track_vars` est activé et `register_globals` est désactivé, seuls les éléments du tableau global `$_SESSION` contiendront les variables enregistrées dans la session. Les variables de sessions relues seront uniquement disponibles dans `$_SESSION`.

L'utilisation de `$_SESSION` (ou `$HTTP_SESSION_VARS` avec PHP 4.0.6 et plus ancien) est recommandé pour une meilleure sécurité et un code plus facilement entretenu. Avec `$_SESSION`, il n'y a pas besoin d'utiliser les fonctions `session_register`, `session_unregister` et `session_is_registered`. Les variables de sessions sont accessibles comme toute autre variable.

Enregistrer une variable avec `$_SESSION`.

```
<?php
session_start();
// Utilisez $HTTP_SESSION_VARS avec PHP 4.0.6 ou plus ancien
if (!isset($_SESSION['compteur'])) {
    $_SESSION['compteur'] = 0;
} else {
    $_SESSION['compteur']++;
}
?>
```

Retirer une variable de session avec `$_SESSION` et `register_globals` inactif.

```
<?php
session_start();
// Utilisez $HTTP_SESSION_VARS avec PHP 4.0.6 ou plus ancien
unset($_SESSION['compteur']);
```

?>

Attention

N'utilisez PAS la fonction `unset` avec `$_SESSION` sous la forme `unset($_SESSION)` sinon, cela rendra impossible l'enregistrement de données dans la session en utilisant la super-globale `$_SESSION`.

Attention

Vous ne pouvez pas utiliser les références sur des variables de session car il n'y a aucune manière faisable de restaurer une référence vers une autre variable.

Si `register_globals` est activé, alors toutes les variables globales peuvent être enregistrées comme variables de session, et toutes les variables de sessions seront reconstituées comme variables globales. Comme PHP doit savoir quels variables globales sont enregistrées comme variables de sessions, l'utilisateur doit enregistrer les variables avec `session_register` tandis que `$HTTP_SESSION_VARS` et `$_SESSION` ne nécessitent pas `session_register`.

Attention

Avant PHP 4.3, si vous utilisez `$_SESSION` et que vous avez désactivé `register_globals`, n'utilisez pas `session_register`, `session_is_registered` ou `session_unregister`.

Désactiver `register_globals` est recommandé pour des raisons de sécurité et de performances.

Si `register_globals` est activé, alors les variables globales et les entrées dans le tableau `$_SESSION` seront des références sur la même valeur pour les valeurs qui auront été enregistrées avant le démarrage de la session (donc, dans les page précédentes). Cependant, si la variable a été enregistrée avec `$_SESSION`, alors, la variable globale est disponible jusqu'à la prochaine requête.

De plus, si vous enregistrez une nouvelle variable avec la fonction `session_register`, l'entrée dans l'environnement global et `$_SESSION` ne fera pas de référence vers la même valeur jusqu'à la prochaine utilisation de `session_start` (ceci s'applique à PHP 4.2 et avant seulement). C'est à dire qu'une modification dans les variables globales ne sera pas répercutée dans les entrées de `$_SESSION`. Il est peu probable que cela ait un impact en pratique et, de plus, cela a été corrigé en PHP 4.3.

8.117.9 Passer l'identifiant de session (session ID)

Il y a deux méthodes de propagation de l'identifiant de session :

- Cookies
- Par URL

Le module de session supporte les deux méthodes. Les cookies sont optimaux, mais comme ils ne sont pas sûrs (tous les internautes ne les acceptent pas), ils ne sont pas fiables. La seconde méthode place l'identifiant de session directement dans les URL.

PHP est capable de faire cela de manière transparente, lorsqu'il est compilé avec l'option `--enable-trans-sid`. Si vous activez cette option, les URL relatives seront modifiées pour contenir l'identifiant de session automatiquement. Alternativement, vous pouvez utiliser la constante `SID`, qui est définie, si le client n'a pas envoyé le cookie approprié. `SID` est soit de la forme `session_name=session_id` ou une chaîne vide.

Note

L'option `arg_separator_output` de `php.ini` vous permet de personnaliser le séparateur d'arguments. Pour être complètement en accord avec les spécifications XHTML, spécifiez `&` ici.

Alternativement, vous pouvez utiliser la constante `SID` qui est définie si la session a commencé. Si le client n'envoie pas un cookie de session approprié, il aura la forme `session_name=session_id`. Sinon, il vaudra une chaîne vide. Ainsi, vous pouvez dans tous les cas l'inclure dans l'URL.

L'exemple suivant vous montre comment enregistrer une variable et comment réaliser un lien correct avec une autre page, avec `SID`.

Compter le nombre de passages d'un utilisateur sur une page

```
<?php
if (!session_is_registered('compteur')) {
    session_register('compteur');
    $compteur = 1;
} else {
    $compteur++;
}
?>

<p>
Bonjour visiteur, vous avez vu cette page <?php echo $compteur; ?> fois.
</p>

<p>
Pour continuer, <a href="nextpage.php?<?php echo strip_tags(SID); ?>">cliquez ici</a>.
</p>
```

La fonction `strip_tags` est utilisée lors de l'affichage du `SID` dans le but de contrer les attaques XSS.

L'affichage du `SID`, comme montré dans l'exemple ci-dessus, n'est pas nécessaire si `--enable-trans-sid` a été utilisé pour compiler PHP.

Note

Les URL non-relatives sont considérées comme externes au site, et ne recevront pas le `SID`, car c'est une fuite d'informations vers un autre site (envoi d'informations importantes).

8.117.10 Gestion personnalisée des sessions

Pour implémenter un stockage en base de données, ou toute autre méthode, vous aurez besoin de la fonction `session_set_save_handler` pour paramétrer vos propres fonctions de stockage.

- [Introduction](#)
- [Sessions et sécurité](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [Passer l'identifiant de session \(session ID\)](#)
- [Gestion personnalisée des sessions](#)
- [session_cache_expire](#)
- [session_cache_limiter](#)
- [session_commit](#)
- [session_decode](#)
- [session_destroy](#)
- [session_encode](#)

- [session_get_cookie_params](#)
- [session_id](#)
- [session_is_registered](#)
- [session_module_name](#)
- [session_name](#)
- [session_regenerate_id](#)
- [session_register](#)
- [session_save_path](#)
- [session_set_cookie_params](#)
- [session_set_save_handler](#)
- [session_start](#)
- [session_unregister](#)
- [session_unset](#)
- [session_write_close](#)

8.117.12 [session_cache_limiter\(\)](#) : Lit et/ou modifie le limiteur de cache de session

string [session_cache_limiter](#) (*string* *cache_limiter*)

[session_cache_limiter](#) retourne la configuration courante du limiteur de cache. Si *cache_limiter* est fourni, le limiteur de cache est reconfiguré avec la valeur de *cache_limiter* .

Le limiteur de cache contrôle les en-têtes HTTP envoyés au client. Certains en-têtes déterminent les règles de mise en cache de la page sur le navigateur. En configurant ce limiteur à *nocache* , par exemple, le navigateur ne mettra pas la page dans son cache. La valeur de *public* , au contraire, permettra le cache. La valeur de *private* est aussi possible, et est légèrement plus restrictive que *public* .

En mode *private* , l'en-tête *Expire* envoyé au client peut poser des problèmes à certains navigateurs, comme, notamment, Mozilla . Vous pouvez éviter ce problème avec le mode *private_no_expire* . L'en-tête *Expire* n'est jamais envoyé au navigateur pour ce mode.

Note

private_no_expire a été ajouté en PHP 4.2.0.

Le limiteur de cache est remis à la valeur par défaut de [session.cache_limiter](#) à chaque démarrage de script PHP. Donc, vous devrez appeler [session_cache_limiter](#) à chaque page, et avant [session_start](#) .

Exemple avec [session_cache_limiter](#)

```
<?php
/* configure le limiteur de cache à 'private' */
session_cache_limiter('private');
$cache_limiter = session_cache_limiter();

echo "Le limiteur de cache vaut maintenant $cache_limiter<br />";
?>
```

Voir aussi la directive de configuration [session.cache_limiter](#) .

8.117.13 `session_commit()` : Alias de `session_write_close`

Cette fonction est un alias de : [session_write_close](#) .

8.117.14 `session_decode()` : Décode les données de session

bool `session_decode` (string data)

`session_decode` décode les données de sessions fournies dans le paramètre data , et définit les variables ainsi contenues dans la session.

Voir aussi [session_encode](#) .

8.117.15 `session_destroy()` : Détruit une session

bool `session_destroy` (void)

`session_destroy` détruit toutes les données associées à la session courante. Cette fonction ne détruit pas les variables globales associées à la session, de même, elle ne détruit pas le cookie de session.

Pour détruire complètement une session, comme faire sortir l'utilisateur, l'identifiant de la session doit également être effacé. Si un cookie est utilisé pour propager l'identifiant de session (comportement par défaut), alors le cookie de session doit être effacé. La fonction [setcookie](#) peut être utilisée pour cela.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Destruction d'une session avec `session_destroy`

```

<?php

// Initialisation de la session.
// Si vous utilisez un autre nom
// session_name("autrenom")
session_start();

// Détruit toutes les variables de session
$_SESSION = array();

// Si vous voulez détruire complètement la session, effacez également
// le cookie de session.
// Note : cela détruira la session et pas seulement les données de session !
if (isset($_COOKIE[session_name()])) {
    setcookie(session_name(), '', time()-42000, '/');
}

// Finalement, on détruit la session.
session_destroy();

?>

```

Note

Utilisez uniquement la fonction [session_unset](#) pour l'ancien code déprécié qui n'utilise pas les variables `$_SESSION` .

Voir aussi [unset](#) et [setcookie](#) .

8.117.16 `session_encode()` : Encode les données de session

string `session_encode` (void)

`session_encode` retourne une chaîne contenant les variables de la session courante encodées et prêtes au stockage.

Voir aussi `session_decode` .

8.117.17 `session_get_cookie_params()` : Lit la configuration du cookie de session

array `session_get_cookie_params` (void)

`session_get_cookie_params` retourne la configuration courante du cookie de session. Cette fonction retourne un tableau, qui contient les éléments suivants :

- " lifetime " - Durée de vie du cookie.
- " path " - Le chemin où les informations sont stockées.
- " domain " - Le domaine du cookie.
- " secure " - Le cookie ne doit être envoyé que sur des connexions sécurisées (cet élément a été ajouté en PHP 4.0.4).

Voir aussi les directives de configuration `session.cookie_lifetime` , `session.cookie_path` , `session.cookie_domain` , `session.cookie_secure` et `session_set_cookie_params` .

8.117.18 `session_id()` : Lit et/ou modifie l'identifiant courant de session

string `session_id` (*string id*)

`session_id` est utilisé pour récupérer ou définir l'identifiant de session pour la session courante.

La constante SID peut également être utilisée pour lire le nom de la session courante et l'identifiant de session à fournir dans les URL. Voyez aussi `Gestion de session` .

8.117.19 `session_is_registered()` : Vérifie si une variable est enregistrée dans la session

bool `session_is_registered` (string name)

`session_is_registered` retourne TRUE si la variable de nom name fait partie de la session courante.

Note
Si <code>\$_SESSION</code> (ou <code>\$HTTP_SESSION_VARS</code> en PHP 4.0.6 et plus ancien) est utilisé, utilisez la fonction <code>isset</code> pour vérifier la présence de la variable dans <code>\$_SESSION</code> .
Attention
Si vous utilisez <code>\$_SESSION</code> (ou <code>\$HTTP_SESSION_VARS</code>), n'utilisez pas <code>session_register</code> , <code>session_is_registered</code> et <code>session_unregister</code> .

8.117.20 `session_module_name()` : Lit et/ou modifie le module de session courant

string `session_module_name` (*string module*)

`session_module_name` retourne le nom du module courant de session. Si module est fourni, cette valeur sera alors utilisée, et remplacera la valeur courante.

8.117.21 `session_name()` : Lit et/ou modifie le nom de la session

string `session_name` (*string name*)

`session_name` retourne le nom de la session courante. Si name est fourni, le nom de la session courante sera remplacé par cette valeur.

Le nom de session est utilisé comme nom de cookie ou nom de variable dans les URL. Il ne doit contenir que des caractères alpha-numériques ; il doit être court et descriptif (surtout pour les utilisateurs ayant activé l'alerte cookie). Le nom de session est toujours remis à sa valeur par défaut (`session.name` , dans `php.ini`). Il faut donc utiliser `session_name` au début de chaque script, et notamment avant `session_start` ou `session_register` .

Attention
Les noms de session ne peuvent contenir que des chiffres, au moins une lettre doit être présente. Sinon, un identifiant de session sera généré à chaque fois.
Exemple avec <code>session_name</code>
<pre><?php /* choisi le nom de session : MaSession */ \$previous_name = session_name("MaSession"); echo "L'ancien nom de la session était \$previous_name
"; ?></pre>

Voir aussi la directive de configuration `session.name` .

8.117.22 `session_regenerate_id()` : Remplace l'identifiant de session courant par un nouveau

bool `session_regenerate_id` (*bool delete_old_session*)

`session_regenerate_id` va remplacer l'identifiant de session courant par un nouveau, généré automatiquement, tout en conservant les valeurs de session.

8.117.23 `session_register()` : Enregistre une variable dans une session

bool `session_register` (*mixed name* , *mixed ...*)

`session_register` enregistre toutes les variables de nom name dans la session courante. Le nombre de variables enregistrées est libre. Les noms peuvent être passés comme des chaînes, ou comme

des tableaux contenant des chaînes ou des tableaux. Pour chaque nom, `session_register` place la variable dans la session courante, pour la sauvegarde de fin de script.

<p>Attention</p> <p>Si vous voulez que votre script fonctionne indépendamment de la configuration de la directive <code>register_globals</code>, vous devez utiliser la variable <code>\$_SESSION</code>. Tous les éléments de <code>\$_SESSION</code> sont automatiquement enregistrés. Si votre script utilise <code>session_register</code>, il ne fonctionnera pas dans les environnements où <code>register_globals</code> est désactivée.</p>
<p>Note</p> <p><code>register_globals</code> : note importante</p> <p>Depuis PHP 4.2.0, la valeur par défaut de la directive de configuration PHP <code>register_globals</code> vaut off. La communauté PHP vous recommande de ne pas dépendre de cette directive, mais de trouver d'autres moyens pour accéder aux données, tels que les <code>superglobals</code>.</p>
<p>Attention</p> <p>Cette fonction enregistre une variable global. Si vous enregistrez une variable globale dans une fonction, vous devez vous assurer de la rendre <u>globale</u>, avec <code>global</code> ou le tableau <code>\$GLOBALS[]</code>, ou bien utiliser un des tableaux de session ci-dessous.</p>
<p>Attention</p> <p>Si vous utilisez <code>\$_SESSION</code> (ou <code>\$HTTP_SESSION_VARS</code>), n'utilisez pas <code>session_register</code>, <code>session_is_registered</code> et <code>session_unregister</code>.</p>

`session_register` retourne TRUE si toutes les variables ont été enregistrées correctement dans la session.

Si `session_start` n'a pas été appelé avant cette fonction, un appel implicite à `session_start` sans aucun paramètre sera fait.

Vous pouvez aussi créer une variable de session, simplement en ajoutant l'index approprié dans la variable `$_SESSION` ou `$HTTP_SESSION_VARS` (PHP < 4.1.0).

<p>Exemple avec <code>session_register</code></p> <pre><?php // L'utilisation de session_register() est déconseillée \$barney = "Un gros dinosaure violet."; session_register("barney"); // L'utilisation de \$_SESSION est encouragée depuis PHP 4.1.0 \$_SESSION["zim"] = "Un envahisseur d'une autre planète."; // L'ancienne méthode avec \$HTTP_SESSION_VARS \$HTTP_SESSION_VARS["spongebob"] = "Il a un caleçon carré !"; ?></pre>
<p>Note</p> <p>Il n'est actuellement pas possible d'enregistrer des ressources dans les sessions. Par exemple, vous ne pouvez pas créer de connexion à une base de données, et stocker la connexion dans une variable de session. Elle ne sera pas valide lors de la prochaine page. Les fonctions PHP qui retournent des ressources sont identifiées avec le type <code>resource</code> dans leurs définitions. Une liste de fonctions qui retournent des ressources sont disponible dans l'annexe types de ressources.</p> <p>Si <code>\$_SESSION</code> (ou <code>\$HTTP_SESSION_VARS</code> pour les versions antérieures à PHP 4.0.6) est utilisé, assignez les variables à <code>\$_SESSION</code> : i.e. <code>\$_SESSION['var'] = 'ABC'</code>;</p>

Voir aussi `session_is_registered`, `session_unregister` et `$_SESSION`.

8.117.24 `session_save_path()` : Lit et/ou modifie le chemin de sauvegarde des sessions

string `session_save_path` (*string path*)

`session_save_path` retourne le chemin du dossier actuellement utilisé pour sauver les données de sessions. Si `path` est spécifié, le chemin du dossier sera modifié. `session_save_path` doit être appelé avant `session_start` .

Note

Sur certains systèmes d'exploitation, vous aurez à choisir un chemin vers un dossier capable de gérer un grand nombre de petits fichiers efficacement. Par exemple, sous Linux, <code>reiserfs</code> peut se rendre plus efficace que <code>ext2fs</code> .
--

Voir aussi la directive de configuration `session.save_path` .

8.117.25 `session_set_cookie_params()` : Modifie les paramètres du cookie de session

void `session_set_cookie_params` (int *lifetime* , *string path* , *string domain* , *bool secure*)

`session_set_cookie_params` modifie les paramètres de configuration du cookie de session, qui a été configuré dans le fichier `php.ini` . L'effet de cette fonction ne dure que pendant l'exécution du script courant. De ce fait, vous devez appeler `session_set_cookie_params` pour chaque script et avant l'appel à `session_start` .

Note

Le paramètre <code>secure</code> a été ajouté en PHP 4.0.4.

Voir aussi les directives de configuration `session.cookie_lifetime` , `session.cookie_path` , `session.cookie_domain` , `session.cookie_secure` et `session_get_cookie_params` .

8.117.26 `session_set_save_handler()` : Configure les fonctions de stockage de sessions

bool `session_set_save_handler` (*string open* , *string close* , *string read* , *string write* , *string destroy* , *string gc*)

`session_set_save_handler` configure les fonctions de stockage de sessions, et permet de choisir des fonctions utilisateurs pour sauver et relire toutes les sessions. Cette fonction est très pratique lorsqu'il faut sauver les données de sessions en utilisant une autre technique que le système par fichier fourni par défaut : notamment, stockage en base de données ou dans des démons tels `SRM` . Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Note

Le gestionnaire d'écriture n'est pas exécuté tant que le flot de sortie est ouvert. Par conséquent, l'affichage de messages de débogage durant le gestionnaire d'écriture ne sera donc jamais visible depuis le navigateur. Si vous avez besoin de débogage, nous vous suggérons d'écrire vos messages dans un fichier.

L'exemple suivant fournit un exemple de stockage des sessions semblable au système par défaut (par fichiers). Cet exemple peut facilement être adapté pour réaliser des sauvegardes en base de données, avec votre serveur préféré.

La fonction de lecture doit toujours retourner une chaîne, pour que le gestionnaire fonctionne comme prévu. Vous devez retourner une chaîne vide s'il n'y a pas de données à lire. Les valeurs retournées par les autres gestionnaires sont converties en booléen avant d'être retournées au script. TRUE pour la réussite, FALSE pour l'échec.

Attention

Les gestionnaires d'écriture et de fermeture sont appelés après la destruction des objets depuis PHP 5.0.5. Ces destructeurs peuvent utiliser les sessions mais le gestionnaire de session ne peut pas utiliser les objets. Dans les versions antérieures, ils étaient appelés dans l'ordre inverse. Il est possible d'appeler `session_write_close` depuis le destructeur pour résoudre ce problème.

Exemple avec `session_set_save_handler`

```
<?php
function open ($save_path, $session_name) {
    global $sess_save_path, $sess_session_name;

    $sess_save_path = $save_path;
    $sess_session_name = $session_name;
    return(true);
}

function close() {
    return(true);
}

function read ($id) {
    global $sess_save_path, $sess_session_name;

    $sess_file = "$sess_save_path/sess_$id";
    if ($fp = @fopen($sess_file, "r")) {
        $sess_data = fread($fp, filesize($sess_file));
        return($sess_data);
    } else {
        return(""); // Doit retourner "" ici.
    }
}

function write ($id, $sess_data) {
    global $sess_save_path, $sess_session_name;

    $sess_file = "$sess_save_path/sess_$id";
    if ($fp = @fopen($sess_file, "w")) {
        return(fwrite($fp, $sess_data));
    } else {
        return(false);
    }
}

function destroy ($id) {
    global $sess_save_path, $sess_session_name;

    $sess_file = "$sess_save_path/sess_$id";
    return(@unlink($sess_file));
}

/*****
 * ATTENTION - Vous devez implémenter un      *
 * collecteur de données obsolescentes ici. *
 *****/
function gc ($maxlifetime) {
    return true;
}
```

```

}

session_set_save_handler ("open", "close", "read", "write", "destroy", "gc");

session_start();

// Utilisez vos sessions comme d'habitude
?>

```

Voir aussi la directive de configuration [session.save_handler](#) .

8.117.27 session_start() : Initialise une session

bool **session_start** (void)

session_start crée une session (ou restaure celle trouvée sur le serveur, via l'identifiant de session passé dans une requête GET, POST ou par un cookie).

session_start retourne toujours TRUE .

Note

Si vous utilisez des sessions basées sur les cookies, vous devez appeler session_start avant d'afficher quoi que ce soit au navigateur.

Exemple de session : page1.php

```

<?php
// page1.php

session_start();

echo 'Bienvenue à la page numéro 1';

$_SESSION['favcolor'] = 'vert';
$_SESSION['animal']    = 'chat';
$_SESSION['time']      = time();

// Fonctionne si le cookie a été accepté
echo '<br /><a href="page2.php">page 2</a>';

// Ou bien, en indiquant explicitement l'identifiant de session
echo '<br /><a href="page2.php?' . SID . '">page 2</a>';
?>

```

Après avoir vu la page page1.php avec un navigateur, la seconde page page2.php va magiquement afficher les données de session. Lisez la référence sur les [sessions](#) pour des informations sur la [propagation des identifiants de session](#) , et l'utilisation de la constante SID .

Exemple de session : page2.php

```

<?php
// page2.php

session_start();

echo 'Bienvenue sur la page numéro 2<br />';

echo $_SESSION['favcolor']; // vert
echo $_SESSION['animal'];  // chat
echo date('Y m d H:i:s', $_SESSION['time']);

```

```
// Vous pourriez utiliser la constante SID ici, tout comme dans la page page1.php
echo '<br /><a href="page1.php">page 1</a>';
?>
```

Si vous avez utilisé une session nommée, vous devez appeler `session_name` avant d'appeler `session_start`.

`session_start` va configurer un gestionnaire interne d'affichage, pour assurer la réécriture des URL, lorsque le support transparent du SID est activé. Si vous utilisez `ob_gzhandler` ou `ob_start`, l'ordre des gestionnaires d'affichage est important pour un affichage correct. Par exemple, l'utilisateur doit enregistrer `ob_gzhandler` avant que la session ne commence.

Note

L'utilisation de `zlib.output_compression` est recommandée, à la place de `ob_gzhandler`.

Note

Depuis PHP 4.3.3, l'appel à `session_start` alors que la session a déjà débuté, retournera une erreur de niveau E_NOTICE. Le second appel à la fonction sera tout simplement ignoré.

Voir aussi `$_SESSION`, `session.auto_start` et `session_id`.

8.117.28 `session_unregister()` : Supprime une variable de la session

bool `session_unregister` (string name)

`session_unregister` supprime la variable name de la session courante.

`session_unregister` retourne TRUE si la variable a pu être supprimée réellement.

Note

Si `$_SESSION` (ou `$HTTP_SESSION_VARS` pour PHP 4.0.6 et plus ancien) est utilisé, utilisez `unset` pour supprimer une variable de session. N'utilisez pas `unset` directement sur `$_SESSION` sinon cela désactivera les fonctions spéciales du tableau superglobal `$_SESSION`.

Attention

`session_unregister` ne supprime pas la variable globale correspondant au nom de name : elle empêche simplement que name soit sauvée de la session à la fin du script. Vous devez appeler `unset` vous-même pour effacer la globale correspondante.

Attention

Si vous utilisez `$_SESSION` (ou `$HTTP_SESSION_VARS`), n'utilisez pas `session_register`, `session_is_registered` et `session_unregister`.

8.117.29 `session_unset()` : Détruit toutes les variables d'une session

void `session_unset` (void)

`session_unset` détruit toutes les variables de la session courante.

Note

Si vous utilisez `$_SESSION` (ou `$HTTP_SESSION_VARS` pour PHP 4.0.6 ou plus ancien), utilisez `unset` pour détruire une variable de session, i.e. `unset($_SESSION['nomvariable']);`.

Attention

Ne détruisez pas <code>\$_SESSION</code> avec <code>unset(\$_SESSION)</code> car cela désactivera la possibilité d'enregistrer des variables de session à partir du tableau superglobal <code>\$_SESSION</code> .

8.117.30 `session_write_close()` : Ecrit les données de session et ferme la session

void **`session_write_close`** (void)

`session_write_close` termine la session courante, après avoir stocké les données.

Les données de session sont généralement stockées à la fin de votre script, automatiquement, sans besoin d'appeler explicitement `session_write_close`. Mais durant toute l'exécution du script, les données de session sont verrouillées en écriture, et un seul script peut opérer sur la session en même temps. Lorsque vous utilisez des frames avec des sessions, vous vous en rendrez compte en voyant les frames se rafraîchir les uns après les autres. Vous pouvez réduire le temps de calcul de ces pages en refermant la session aussitôt que possible, ce qui libère les données pour les autres scripts.

8.118 Mémoire partagée

8.118.1 Introduction

Shmop est un ensemble de fonctions simples pour gérer la mémoire partagée avec PHP (lecture, écriture, création et suppression de segments de mémoire partagée UNIX).

Note

Les versions de Windows plus anciennes que Windows 2000 ne supportent pas la mémoire partagée. Sous Windows, Shmop ne fonctionnera que si PHP fonctionne comme module de serveur web, comme Apache ou IIS (CLI et CGI ne fonctionneront pas).

Note

En PHP 4.0.3, ces fonctions étaient préfixées avec shm au lieu de shmop .

8.118.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.118.3 Installation

Pour utiliser les fonctions shmop, compilez PHP avec l'option `--enable-shmop` .

8.118.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.118.5 Types de ressources

8.118.6 Constantes pré-définies

Cette extension ne définit aucune constante.

8.118.7 Exemples

Introduction à la mémoire partagée

```
<?php
// Crée 100 octets de mémoire partagée avec
// un identifiant système "0xff3"
$shm_id = shmop_open(0xff3, "c", 0644, 100);
if (!$shm_id) {
    echo "Impossible de créer la mémoire partagée\n";
}

// Lire la taille de la mémoire partagée
$shm_size = shmop_size($shm_id);
echo "Un bloc de SHM de taille " . $shm_size . " a été créé.\n";
```

```

// Ecriture d'une chaîne de test dans ce segment
$shm_bytes_written = shmop_write($shm_id, "Mon bloc de mémoire partagée", 0);
if ($shm_bytes_written != strlen("Mon bloc de mémoire partagée")) {
    echo "Impossible d'écrire toutes les données en mémoire\n";
}

// Lecture du segment
$my_string = shmop_read($shm_id, 0, $shm_size);
if(!$my_string) {
    echo "Impossible de lire toutes les données en mémoire\n";
}
echo "Les données mises en mémoire partagées sont : " . $my_string . "\n";

// Maintenant, effaçons le bloc, et fermons le segment de mémoire
if(!shmop_delete($shm_id)) {
    echo "Impossible d'effacer le segment de mémoire";
}
shmop_close($shm_id);

?>

```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [shmop_close](#)
- [shmop_delete](#)
- [shmop_open](#)
- [shmop_read](#)
- [shmop_size](#)
- [shmop_write](#)

8.118.9 shmop_delete() : Détruit un bloc de mémoire partagée

bool **shmop_delete** (int shm_id)

shmop_delete sert à détruire un bloc de mémoire partagée.

shmop_delete prend un identifiant de mémoire partagée shm_id , créé par shmop_open . En cas de succès, la fonction retourne 1, et sinon, 0.

Effacement d'un bloc de mémoire partagée

```

<?php
shmop_delete($shm_id);
?>

```

Ce exemple efface le bloc de mémoire partagée identifié par \$shm_id .

8.118.10 shmop_open() : Crée ou ouvre un bloc de mémoire partagée

int **shmop_open** (int key , string flags , int mode , int size)

shmop_open peut créer ou ouvrir un bloc de mémoire partagée.

shmop_open prend 4 paramètres : la clé, qui sera l'identifiant système pour le bloc ; ce paramètre peut être passé comme un décimal ou un hexadécimal. Le deuxième paramètre est un groupe d'options :

- "a" pour accès (utilise SHM_RDONLY pour shmat) utilisez cette option pour ouvrir un bloc déjà existant en lecture seule.
- "c" pour création (utilise IPC_CREATE) utilisez cette option pour créer un nouveau bloc, ou, si un segment avec le même identifiant existe, essayer d'y accéder en lecture et écriture.
- "w" pour accès en lecture et écriture. Utilisez cette option lorsque vous devez accéder en lecture et écriture à un segment de mémoire partagée. C'est le cas le plus courant.
- "n" crée un nouveau segment de mémoire partagée (utilise IPC_CREATE|IPC_EXCL). Utilisez cette option lorsque vous voulez créer un nouveau segment de mémoire partagée sauf s'il en existe déjà un corrompu avec la même option. Ceci est très pratique pour des raisons de sécurité, pour éviter des trous de sécurité qui exploiteraient la course aux ressources.

Le troisième paramètre est le mode, c'est à dire les permissions que vous donnez à ce bloc. Ce sont les mêmes que pour les fichiers. Ces permissions doivent être passées sous forme d'octal (i.e. 0644). Le dernier paramètre est la taille du bloc de mémoire, en octets.

Note

Les troisième et quatrième paramètres doivent être passés à 0 si vous voulez ouvrir un bloc de mémoire partagée déjà existant. En cas de succès shmop_open retourne un identifiant que vous pouvez utiliser pour accéder à la mémoire que vous venez de créer.

Créer un nouveau bloc de mémoire partagée Shmop

```
<?php
$shm_key = ftok(__FILE__, 't');
$shm_id = shmop_open($shm_key, "c", 0644, 100);
?>
```

Cet exemple ouvre un nouveau bloc de mémoire partagée, dont l'identifiant est retourné par ftok .

8.118.11 shmop_read() : Lit des données à partir d'un bloc

string **shmop_read** (int shmid , int start , int count)

shmop_read lit une chaîne dans une bloc de mémoire partagée.

shmop_read prend 3 paramètres: shmid , qui est un identifiant de mémoire partagée, créé par shmop_open , start qui est la position à partir de laquelle on commence à lire dans la mémoire et count , le nombre d'octets à lire.

Lire un bloc de mémoire partagée

```
<?php
$shm_data = shmop_read($shm_id, 0, 50);
?>
```

Cet exemple lit 50 octets dans le bloc de mémoire partagée et les place dans `$shm_data` .

8.118.12 `shmop_size()` : Lire la taille du bloc de mémoire partagée

int `shmop_size` (int `shmid`)

`shmop_size` sert à connaître la taille en octets d'un bloc de mémoire partagée.

`shmop_size` prend comme argument `shmid` , un identifiant de bloc de mémoire partagée créé par `shmop_open` , et retourne un entier, qui représente le taille de ce bloc.

Lire la taille d'un bloc de mémoire partagée

```
<?php
$shm_size = shmop_size($shm_id);
?>
```

Cet exemple lit la taille du bloc identifié par `$shm_id` , et la place dans `$shm_size` .

8.118.13 `shmop_write()` : Ecrire dans un bloc de mémoire partagée

int `shmop_write` (int `shmid` , string `data` , int `offset`)

`shmop_write` écrit une chaîne dans un bloc de mémoire partagée.

`shmop_write` prend 3 paramètres: `shmid` , qui est un identifiant de mémoire partagée, créé par `shmop_open` , `data` qui est la chaîne à écrire dans la mémoire et `offset` , la position à partir de laquelle il faut commencer à écrire.

Ecrire un bloc de mémoire partagée

```
<?php
$shm_bytes_written = shmop_write($shm_id, $my_string, 0);
?>
```

Cet exemple écrit les données de la chaîne `$my_string` dans un bloc de mémoire partagée. `$shm_bytes_written` représentera le nombre d'octets écrits.

8.119 Fonctions SimpleXML

8.119.1 Introduction

L'extension SimpleXML fournit des outils très simples et faciles à utiliser pour convertir du XML en un objet qui peut être manipulé avec ses propriétés et les itérateurs de tableaux.

8.119.2 Pré-requis

L'extension SimpleXML requiert PHP 5.

8.119.3 Installation

L'extension SimpleXML est activé par défaut. Pour la désactiver, utilisez l'option de configuration `--disable-simplexml` .

8.119.4 Exemples

Plusieurs exemples de ce chapitre requièrent une chaîne XML. Plutôt que de la répéter à chaque exemple, nous allons la placer dans un fichier que nous incluerons dans chacun d'entre eux. Le contenu de ce fichier est illustré par l'exemple qui suit. Autrement, vous pouvez créer un document XML et le lire avec `simplexml_load_file` .

Fichier avec une chaîne XML qui sera inclus partout : `example.php`

```
<?php
$xmlstr = <<<XML

<movies>
<movie>
  <title>PHP: Behind the Parser</title>
  <characters>
    <character>
      <name>Ms. Coder</name>
      <actor>Onlvivia Actora</actor>
    </character>
    <character>
      <name>Mr. Coder</name>
      <actor>El Act&oacute;r</actor>
    </character>
  </characters>
  <plot>
    So, this language. It's like, a programming language. Or is it a
    scripting language? All is revealed in this thrilling horror spoof
    of a documentary.
  </plot>
  <rating type="thumbs">7</rating>
  <rating type="stars">5</rating>
</movie>
</movies>
XML;
?>
```

La simplicité de SimpleXML apparaît plus clairement lorsqu'on essaye d'extraire une chaîne ou un nombre d'un document XML basique.

Obtenir <plot>

```
<?php
include 'example.php';

$xml = simplexml_load_string($xmlstr);

echo $xml->movie[0]->plot; // "So this language. It's like..."
?>
```

Accéder à un élément non-unique avec SimpleXML

Lorsque plusieurs instances d'un élément existent en tant que fils d'un élément père unique, les techniques normales d'itération peuvent être appliquées.

```
<?php
include 'example.php';

$xml = simplexml_load_string($xmlstr);

/* Pour chaque <movie>, nous affichons un <plot>. */
foreach ($xml->movie as $movie) {
    echo $movie->plot, '<br />';
}

?>
```

Utilisation des attributs

Jusque là, nous n'avons couvert que la lecture des noms d'éléments et leurs valeurs. SimpleXML peut aussi atteindre leurs attributs. L'accès aux attributs d'un élément se fait de la même façon que l'accès aux éléments d'un tableau.

```
<?php
include 'example.php';

$xml = simplexml_load_string($xmlstr);

/* Accès au node <rating> du premier <movie>.
 * Affichage des attributs de <rating> également. */
foreach ($xml->movie[0]->rating as $rating) {
    switch((string) $rating['type']) { // Récupération des attributs comme indices d'éléments
        case 'thumbs':
            echo $rating, ' thumbs up';
            break;
        case 'stars':
            echo $rating, ' stars';
            break;
    }
}

?>
```

Comparaison des éléments et des attributs avec du texte

Pour comparer un élément ou un attribut avec une chaîne de caractères ou pour le passer à une fonction qui nécessite une chaîne de caractères, vous devez le transtiper en une chaîne en utilisant (string) . Sinon, PHP traitera l'élément comme un objet.

```
<?php
include 'example.php';

$xml = simplexml_load_string($xmlstr);
```

```

if ((string) $xml->movie->title == 'PHP: Behind the Parser') {
    echo 'My favorite movie.';
}

htmlentities((string) $xml->movie->title);
?>

```

Utilisation de Xpath

SimpleXML inclut le support embarqué de XPATH . Pour trouver tous les éléments <character> :

```

<?php
include 'example.php';
$xml = simplexml_load_string($xmlstr);

foreach ($xml->xpath('//character') as $character) {
    echo $character->name, 'played by ', $character->actor, '<br />';
}
?>

```

'// ' sert de joker. Pour spécifier un chemin absolu, enlevez un slash.

Attribuer des valeurs

Les données dans SimpleXML n'ont pas à être constantes. L'objet permet la manipulation de tous ces éléments.

```

<?php
include 'example.php';
$xml = simplexml_load_string($xmlstr);

$xml->movie[0]->characters->character[0]->name = 'Miss Coder';

echo $xml->asXML();
?>

```

L'exemple précédent affichera un nouveau document XML, tout comme l'original mise à part que le nouveau document XML définit Ms. Coder à Miss Coder .

Interopérabilité DOM

PHP possède un mécanisme pour convertir les nodes XML entre les formats SimpleXML et DOM. Cet exemple montre comment changer un élément DOM en SimpleXML.

```

<?php
$dom = new domDocument;
$dom->loadXML('<books><book><title>blah</title></book></books>');
if (!$dom) {
    echo 'Error while parsing the document';
    exit;
}

$s = simplexml_import_dom($dom);

echo $s->book[0]->title;
?>

```

- [Introduction](#)

- [Pré-requis](#)
- [Installation](#)
- [Exemples](#)
- [SimpleXMLElement->asXML](#)
- [SimpleXMLElement->attributes](#)
- [SimpleXMLElement->children](#)
- [SimpleXMLElement->xpath](#)
- [simplexml_import_dom](#)
- [simplexml_load_file](#)
- [simplexml_load_string](#)

8.119.6 SimpleXMLElement->attributes() : Identifie les attributs d'un élément

SimpleXMLElement **simplexml_element->attributes** (*string data*)

[simplexml_element->attributes](#) fournit les attributs et les valeurs définies dans une balise XML.

Note

SimpleXML ajoute des propriétés itératives pour presque toutes ses méthodes. Celles-ci ne peuvent être vues en utilisant [var_dump](#) ou tout autre fonction qui examine les objets.

Interprétation d'une chaîne XML

```
<?php
$string = <<<XML
<a>
  <foo name="one" game="lonely">1</foo>
</a>
XML;

$xml = simplexml_load_string($string);
foreach($xml->foo[0]->attributes() as $a => $b) {
    echo $a, '="'', $b, "\"\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
name="one"
game="lonely"
```

8.119.7 SimpleXMLElement->children() : Trouve les enfants d'une node

SimpleXMLElement **simplexml_element->children** (*string namespace*)

Cette méthode trouve le fils de l'élément dont il est membre. Le résultat suit les règles normales d'itération.

Note

SimpleXML ajoute des propriétés itératives pour presque toutes ses méthodes. Celles-ci ne peuvent être vues en utilisant [var_dump](#) ou tout autre fonction qui examine les objets.

Traverser un pseudo-tableau [children\(\)](#)

```

<?php
$xml = simplexml_load_string(
'<person>
  <child role="son">
    <child role="daughter"/>
  </child>
  <child role="daughter">
    <child role="son">
      <child role="son"/>
    </child>
  </child>
</person>');

foreach ($xml->children() as $second_gen) {
  echo ' The person begot a ' . $second_gen['role'];

  foreach ($second_gen->children() as $third_gen) {
    echo ' who begot a ' . $third_gen['role'] . ' ';

    foreach ($third_gen->children() as $fourth_gen) {
      echo ' and that ' . $third_gen['role'] .
        ' begot a ' . $fourth_gen['role'];
    }
  }
}
?>

```

L'exemple ci-dessus va afficher :

The person begot a son who begot a daughter; The person
begot a daughter who begot a son; and that son begot a son

8.119.8 SimpleXMLElement->xpath() : Exécute une requête Xpath sur des données XML

array SimpleXMLElement->xpath (string path)

La méthode xpath cherche dans la node SimpleXML des enfants qui correspondent au path XPATH . Elle retourne toujours un tableau d'objets array SimpleXMLElement.

Xpath

```

<?php
$string = <<<XML
<a>
  <b>
    <c>text</c>
    <c>stuff</c>
  </b>
  <d>
    <c>code</c>
  </d>
</a>
XML;

$xml = simplexml_load_string($string);

/* On cherche <a><b><c> */
$result = $xml->xpath('/a/b/c');

while(list( , $node) = each($result)) {

```

```

    echo '/a/b/c: ', $node, "\n";
}

/* Les chemins relatifs fonctionnent aussi... */
$result = $xml->xpath('b/c');

while(list( , $node) = each($result)) {
    echo 'b/c: ', $node, "\n";
}
?>

```

L'exemple ci-dessus va afficher :

```

/a/b/c: text
/a/b/c: stuff
b/c: text
b/c: stuff

```

Notez que les deux résultats sont égaux.

8.119.9 `simplexml_import_dom()` : Construit un objet SimpleXMLElement à partir d'une node DOM

SimpleXMLElement `simplexml_import_dom` (DOMNode `node` , *string* `class_name`)

`simplexml_import_dom` prend une node d'un document DOM et la transforme en node SimpleXML. Ce nouvel objet peut alors être utilisé comme un élément natif SimpleXML. Si une erreur survient, la fonction retourne FALSE .

Import DOM

```

<?php
$dom = new domDocument;
$dom->loadXML('<books><book><title>blah</title></book></books>');
if (!$dom) {
    echo "Erreur durant l'analyse du document";
    exit;
}

$s = simplexml_import_dom($dom);

echo $s->book[0]->title; // blah
?>

```

Voir aussi [dom_import_simplexml](#) .

8.119.10 `simplexml_load_file()` : Convertit un fichier XML en objet

objet SimpleXMLElement `simplexml_load_file` (*string* `filename` , *string* `class_name` , *int* `options`)

`simplexml_load_file` convertit le document XML `filename` en un objet de type SimpleXMLElement . Si une erreur survient lors de l'accès au fichier ou de son interprétation, cette fonction retourne FALSE .

Vous pouvez utiliser le paramètre optionnel `class_name` et ainsi, la fonction `simplexml_load_file`

retournera un objet de la classe spécifiée. Cette classe doit étendre la classe SimpleXMLElement .

Depuis PHP 5.1.0 et Libxml 2.6.0, vous pouvez aussi utiliser le paramètre options pour spécifier des paramètres additionnels Libxml .

Note

Libxml 2 supprime la protection des caractères des URI, alors si vous voulez passer par exemple b&c comme paramètre URI à a , vous devez appeler `simplexml_load_file(rawurlencode('http://example.com/?a=' . urlencode('b&c')))` . Depuis PHP 5.1.0, vous n'avez plus besoin de faire cela puisque PHP le fait pour vous.

Interprétation d'un document XML

```
<?php
// Le fichier test.xml contient un document XML avec un élément racine
// et au moins un élément /[racine]/title.

if (file_exists('test.xml')) {
    $xml = simplexml_load_file('test.xml');

    var_dump($xml);
} else {
    exit('Echec lors de l'ouverture du fichier test.xml.');
```

L'exemple ci-dessus va afficher :

```
SimpleXMLElement Object
(
    [title] => Titre de l'exemple
    ...
)
```

A partir de là, vous pouvez utiliser `$xml->title` et tout autre élément.

Voir aussi [simplexml_load_string](#) .

8.119.11 simplexml_load_string() : Convertit une chaîne XML en objet

objet `simplexml_load_string` (string data , string class_name , int options)

`simplexml_load_string` convertit la chaîne XML data et retourne un objet de la classe SimpleXMLElement possédant des propriétés contenant les données de la chaîne. En cas d'erreur, cette fonction retourne FALSE .

Vous pouvez utiliser le paramètre optionnel class_name et, ainsi, la fonction `simplexml_load_string` retournera un objet de la classe spécifiée. Cette classe doit étendre la classe SimpleXMLElement .

Depuis PHP 5.1.0 et Libxml 2.6.0, vous pouvez aussi utiliser le paramètre options pour spécifier des paramètres additionnels Libxml .

Convertir une chaîne XML

```
<?php
$string = <<<XML
```

```
<document>
<title>Forty What?</title>
<from>Joe</from>
<to>Jane</to>
<body>
  I know that's the answer -- but what's the question?
</body>
</document>
XML;

$xml = simplexml_load_string($string);

var_dump($xml);
?>
```

L'exemple ci-dessus va afficher :

```
SimpleXMLElement Object
(
  [title] => Forty What?
  [from] => Joe
  [to] => Jane
  [body] =>
  I know that's the answer -- but what's the question?
)
```

A partir de là, vous pouvez utiliser `$xml->body` et tout autre élément.

Voir aussi [simplexml_load_file](#) .

8.120 Fonctions SNMP

8.120.1 Introduction

8.120.2 Pré-requis

Afin de pouvoir utiliser les fonctions SNMP sous Unix, vous aurez besoin d'installer le package NET-SNMP . Sous Windows ces fonctions ne sont disponibles que sous NT, et pas sous Win95/98.

8.120.3 Installation

Important : afin d'utiliser le package UCD SNMP , vous devez mettre la variable NO_ZEROLENGTH_COMMUNITY à 1 avant de compiler. Après avoir configuré UCD SNMP , éditez le fichier config.h et recherchez la valeur NO_ZEROLENGTH_COMMUNITY . Décommentez la ligne avec le #define. Cela doit ressembler à ceci :

```
#define NO_ZEROLENGTH_COMMUNITY 1
```

Ensuite, compilez PHP avec l'option --with-snmp[=DIR] .

Si vous rencontrez des erreurs étranges de "segmentation fault", lors de l'utilisation des commandes SNMP , c'est que vous n'avez pas suivi les recommandations précédentes. Si vous ne voulez pas recompiler UCD SNMP , vous pouvez aussi recompiler PHP avec l'option --enable-ucd-snmp-hack qui évitera cette erreur.

La distribution Windows contient des fichiers nécessaires à SNMP dans le dossier mibs . Ce dossier doit être déplacé dans DRIVE:\usr\mibs , où DRIVE doit être remplacé par la lettre de driver où PHP est installé, c'est-à-dire, par exemple c:\usr\mibs .

8.120.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.120.5 Types de ressources

8.120.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

- SNMP_VALUE_LIBRARY (entier)
- SNMP_VALUE_PLAIN (entier)
- SNMP_VALUE_OBJECT (entier)
- SNMP_BIT_STR (entier)
- SNMP_OCTET_STR (entier)
- SNMP_OPAQUE (entier)

SNMP_NULL ([entier](#))
SNMP_OBJECT_ID ([entier](#))
SNMP_IPADDRESS ([entier](#))
SNMP_COUNTER ([entier](#))
SNMP_UNSIGNED ([entier](#))
SNMP_TIMETICKS ([entier](#))
SNMP_UINTEGER ([entier](#))
SNMP_INTEGER ([entier](#))
SNMP_COUNTER64 ([entier](#))

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [snmp_get_quick_print](#)
- [snmp_get_valueretrieval](#)
- [snmp_read_mib](#)
- [snmp_set_enum_print](#)
- [snmp_set_oid_numeric_print](#)
- [snmp_set_quick_print](#)
- [snmp_set_valueretrieval](#)
- [snmpget](#)
- [snmpgetnext](#)
- [snmprealwalk](#)
- [snmpset](#)
- [snmpwalk](#)
- [snmpwalkoid](#)

8.120.8 [snmp_get_valueretrieval\(\)](#) : Retourne la méthode avec laquelle les valeurs SNMP seront retournées

int [snmp_get_valueretrieval](#) (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.120.9 [snmp_read_mib\(\)](#) : Lit et analyse un fichier MIB dans l'arbre actif MIB

bool [snmp_read_mib](#) (string filename)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.120.10 [snmp_set_enum_print\(\)](#) : Retourne toutes les valeurs qui sont des énumérations avec leur valeur d'énumération au lieu de l'entier

void [snmp_set_enum_print](#) (int enum_print)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.120.11 `snmp_set_oid_numeric_print()` : Retourne tous les objets y compris leur identifiant d'objet dans celui spécifié

```
void snmp_set_oid_numeric_print ( int oid_numeric_print )
```

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.120.12 `snmp_set_quick_print()` : Ecrit la valeur courante de l'option `quick_print` de la bibliothèque UCD

```
void snmp_set_quick_print ( bool quick_print )
```

`snmp_set_quick_print` fixe la valeur de l'option `quick_print` de la bibliothèque UCD SNMP . Lorsqu'elle a la valeur de (1), la bibliothèque SNMP retournera des valeurs 'rapides'. Cela signifie que seule, la valeur sera retournée. Lorsqu'elle a la valeur de (0), la bibliothèque va afficher d'autres informations (telles que l'adresse IP (IpAddress) ou OID). De plus, si `quick_print` n'est pas activée, la bibliothèque affichera aussi des valeurs hexadécimales supplémentaires pour toutes les chaînes de trois caractères ou moins.

Modifier `quick_print` est plus fréquent lorsqu'on utilise les valeurs retournées que lorsqu'on les affiche.

Exemple avec <code>snmp_set_quick_print</code>
--

```
<?php
snmp_set_quick_print(0);
$a = snmpget("127.0.0.1", "public", ".1.3.6.1.2.1.2.2.1.9.1");
echo "$a<br />\n";
snmp_set_quick_print(1);
$a = snmpget("127.0.0.1", "public", ".1.3.6.1.2.1.2.2.1.9.1");
echo "$a<br />\n";
?>
```

La première valeur affichée sera : 'Timeticks: (0) 0:00:00.00', tandis qu'avec `quick_print` activée, seul '0:00:00.00' sera affiché.

Par défaut, UCD SNMP retourne des valeurs détaillées, et `quick_print` sert à ne retourner que la valeur.

Actuellement, les chaînes sont toujours retournées avec des guillemets supplémentaires. Ceci sera corrigé ultérieurement.

`snmp_set_quick_print` ne fonctionne qu'avec la bibliothèque UCD SNMP . `snmp_set_quick_print` n'est pas disponible avec la bibliothèque Windows SNMP .

8.120.13 `snmp_set_valueretrieval()` : Spécifie la méthode avec laquelle les valeurs SNMP seront retournées

```
void snmp_set_valueretrieval ( int method )
```

8.120.10 `snmp_set_enum_print()` : Retourne toutes les valeurs qui sont des énumérations avec leur valeur

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.120.14 snmpget() : Reçoit un objet SNMP

string **snmpget** (string hostname , string community , string object_id , int timeout , int retries)

snmpget retourne un objet SNMP en cas de succès, et FALSE en cas d'erreur.

snmpget sert à lire une valeur d'un objet SNMP représenté par object_id . L'agent SNMP est défini par hostname et la communauté de lecture est spécifiée par le paramètre community .

Exemple avec <u>snmpget</u>

<pre><?php \$syscontact = snmpget("127.0.0.1", "public", "system.SysContact.0"); ?></pre>

8.120.15 snmpgetnext() : Retourne un objet SNMP

string **snmpgetnext** (string host , string community , string object_id , int timeout , int retries)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.120.16 snmprealwalk() : Retourne tous les objets, y compris leur ID d'objet

array **snmprealwalk** (string host , string community , string object_id , int timeout , int retries)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.120.17 snmpset() : Configure un objet SNMP

bool **snmpset** (string hostname , string community , string object_id , string type , mixed value , int timeout , int retries)

snmpset modifie la valeur de l'objet SNMP spécifié, en retournant TRUE en cas de succès et FALSE en cas d'erreur.

snmpset sert à affecter une valeur donnée à un objet SNMP , référencé par object_id . L'agent SNMP est défini par hostname et la communauté de lecture est spécifiée par le paramètre community .

8.120.18 snmpwalk() : Reçoit tous les objets SNMP d'un agent

array **snmpwalk** (string hostname , string community , string object_id , int timeout , int retries)

snmpwalk retourne un tableau d'objets SNMP , en commençant à partir de object_id comme racine, ou FALSE en cas d'erreur.

snmpwalk sert à lire toutes les valeurs d'un agent SNMP , défini par hostname . community définit la communauté de lecture de l'agent. Un objet (object_id = NULL) sert de racine à l'arbre d'objet SNMP et tous les objets sous cette racine sont retournés dans un tableau. Si object_id est spécifié, tous les objets SNMP sous cet objet sont retournés.

Lire tous les objets avec snmpwalk

```
<?php
$a = snmpwalk("127.0.0.1", "public", "");
?>
```

La fonction ci-dessus va retourner tous les objets SNMP d'un agent SNMP qui fonctionnerait sur l'hôte local (localhost). Il suffit alors de faire une boucle pour travailler avec chacun des objets.

Afficher tous les objet avec snmpwalk

```
<?php
foreach ($a as $val) {
    echo "$val\n";
}
?>
```

8.120.19 snmpwalkoid() : Demande d'informations d'arbre sur une entité du réseau

array **snmpwalkoid** (string hostname , string community , string object_id , int timeout , int retries)

snmpwalkoid retourne un tableau associatif, avec les identifiants d'objet et les objets associés, pour tous les objets situés sous la racine object_id , ou FALSE en cas d'erreur.

snmpwalkoid sert à lire tous les identifiants d'objet, et leur valeurs respectives, depuis un serveur SNMP . community indique la communauté de lecture pour cet agent. Un object_id NULL signifie qu'il faut utiliser la racine de l'arbre SNMP et tous les objets sous cet arbre seront retournés. Si object_id est spécifié, tous les objets SNMP situés sous cet objet seront retournés.

La fonction ci-dessous va lire tous les objets de l'agent SNMP qui fonctionne sur l'hôte local. Il est alors possible de les passer en revue avec une boucle : l'existence de snmpwalkoid et snmpwalk est une question d'évolution. Ces deux fonctions sont fournies pour des raisons de compatibilité ascendante.

Lire tous les objets avec snmpwalkoid

```
<?php
$a = snmpwalkoid("127.0.0.1", "public", "");
?>
```

La fonction ci-dessous va lire tous les objets de l'agent SNMP qui fonctionne sur l'hôte local. Il est alors possible de les passer en revue avec une boucle :

Lire tous les objet avec `snmpwalkoid` (2)

```
<?php
for (reset($a); $i = key($a); next($a)) {
    echo "$i: $a[$i]<br />\n";
}
?>
```

8.121 Fonctions SOAP

8.121.1 Introduction

L'extension SOAP sert à écrire des serveurs et clients SOAP. Elle supporte une partie des spécifications de [SOAP 1.1](#) , [SOAP 1.2](#) et [WSDL 1.1](#) .

8.121.2 Pré-requis

Cette extension nécessite la bibliothèque [GNOME_xml](#) . Téléchargez et installez cette bibliothèque. Vous devez avoir au moins la version libxml-2.5.4.

8.121.3 Installation

Cette extension est uniquement disponible si PHP a été configuré avec `--enable-soap` .

8.121.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable	Historique
<code>soap.wsdl_cache_enabled</code>	"1"	PHP_INI_ALL	Disponible depuis PHP 5.0.0.
<code>soap.wsdl_cache_dir</code>	"/tmp"	PHP_INI_ALL	Disponible depuis PHP 5.0.0.
<code>soap.wsdl_cache_ttl</code>	86400	PHP_INI_ALL	Disponible depuis PHP 5.0.0.

Pour plus de détails sur les constantes `PHP_INI_*` , reportez-vous à [Directives du php.ini](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

- `soap.wsdl_cache_enabled` boolean
Active ou désactive le cache WSDL.
- `soap.wsdl_cache_dir` string
Spécifie le nom du dossier où SOAP place les fichiers mis en cache.
- `soap.wsdl_cache_ttl` int
Nombre de secondes de conservation (time to live) des fichiers dans le système de cache.

8.121.5 Classes pré-définies

8.121.5.1 SoapClient

8.121.5.1.1 Constructeur

- `SoapClient->__construct()` - construit un nouvel objet SoapClient

8.121.5.1.2 Méthodes

- SoapClient-> call() - Appelle une fonction SOAP (dépréciée)
- SoapClient-> doRequest() - Exécute une requête SOAP
- SoapClient-> getFunctions() - Retourne une liste de fonctions SOAP
- SoapClient-> getLastRequest() - Retourne la dernière requête SOAP
- SoapClient-> getLastRequestHeaders() - Retourne les en-têtes de la dernière requête SOAP
- SoapClient-> getLastResponse() - Retourne la dernière réponse SOAP
- SoapClient-> getLastResponseHeaders() - Retourne les en-têtes de la dernière réponse SOAP
- SoapClient-> getTypes() - Retourne une liste des types SOAP
- SoapClient-> setCookie() - Définit le cookie qui sera envoyé avec la requête SOAP
- SoapClient-> soapCall() - Appel une fonction SOAP

8.121.5.2 SoapFault

8.121.5.2.1 Constructeur

- SoapFault-> __construct() - construit un nouvel objet SoapFault

8.121.5.3 SoapHeader

SoapHeader est une classe de bas niveau pour passer et retourner des en-têtes SOAP. C'est juste un gestionnaire de données et il n'a pas de méthodes spéciales excepté son constructeur. Elle peut être utilisée dans la méthode SoapClient-> soapCall() pour passer un en-tête SOAP ou dans un gestionnaire d'en-têtes SOAP pour retourner l'en-tête dans une réponse SOAP.

8.121.5.3.1 Constructeur

- SoapHeader-> __construct() - construit un nouvel objet SoapHeader

8.121.5.4 SoapParam

SoapParam est une classe bas niveau pour nommer les paramètres et retourner les valeurs dans le mode non-WSDL . C'est juste un gestionnaire de données et il n'a pas de méthodes spéciales excepté son constructeur.

8.121.5.4.1 Constructeur

- SoapParam-> __construct() - construit un nouvel objet SoapParam

8.121.5.5 SoapServer

8.121.5.5.1 Constructeur

- SoapServer->__construct() - construit un nouvel objet SoapServer

8.121.5.5.2 Méthodes

- SoapServer->addFunction() - Ajoute une ou plusieurs fonctions qui seront gérées dans les requêtes SOAP
- SoapServer->fault() -
- SoapServer->getFunctions() - Retourne une liste de fonctions définies
- SoapServer->handle() - Gère une requête SOAP
- SoapServer->setClass() - Définit une classe qui sera gérée dans les requêtes SOAP
- SoapServer->setPersistence() - Définit le mode persistant de SoapServer

8.121.5.6 SoapVar

SoapVar est une classe bas niveau pour encoder les paramètres et retourner les valeurs dans le mode non-WSDL . C'est juste un gestionnaire de données et n'a pas de méthodes spaciales excepté le constructeur. C'est particulièrement utile lorsque vous voulez définir la propriété type dans une requête ou une réponse SOAP.

8.121.5.6.1 Constructeur

- SoapVar->__construct() - construit un nouvel objet SoapVar

8.121.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

SOAP_1_1 (entier)
SOAP_1_2 (entier)
SOAP_PERSISTENCE_SESSION (entier)
SOAP_PERSISTENCE_REQUEST (entier)
SOAP_FUNCTIONS_ALL (entier)
SOAP_ENCODED (entier)
SOAP_LITERAL (entier)
SOAP_RPC (entier)
SOAP_DOCUMENT (entier)
SOAP_ACTOR_NEXT (entier)
SOAP_ACTOR_NONE (entier)
SOAP_ACTOR_UNLIMITED (entier)
SOAP_COMPRESSION_ACCEPT (entier)
SOAP_COMPRESSION_GZIP (entier)
SOAP_COMPRESSION_DEFLATE (entier)
UNKNOWN_TYPE (entier)
XSD_STRING (entier)
XSD_BOOLEAN (entier)
XSD_DECIMAL (entier)
XSD_FLOAT (entier)
XSD_DOUBLE (entier)

XSD_DURATION (entier)
XSD_DATETIME (entier)
XSD_TIME (entier)
XSD_DATE (entier)
XSD_GYEARMONTH (entier)
integer XSD_GYEAR (entier)
XSD_GMONTHDAY (entier)
XSD_GDAY (entier)
XSD_GMONTH (entier)
XSD_HEXBINARY (entier)
XSD_BASE64BINARY (entier)
XSD_ANYURI (entier)
XSD_ANYXML (integer)
Ajouté en PHP 5.1.0.
XSD_QNAME (entier)
XSD_NOTATION (entier)
XSD_NORMALIZEDSTRING (entier)
XSD_TOKEN (entier)
XSD_LANGUAGE (entier)
XSD_NMTOKEN (entier)
XSD_NAME (entier)
XSD_NCNAME (entier)
XSD_ID (entier)
XSD_IDREF (entier)
XSD_IDREFS (entier)
XSD_ENTITY (entier)
XSD_ENTITIES (entier)
XSD_INTEGER (entier)
XSD_NONPOSITIVEINTEGER (entier)
XSD_NEGATIVEINTEGER (entier)
XSD_LONG (entier)
XSD_INT (entier)
XSD_SHORT (entier)
XSD_BYTE (entier)
XSD_NONNEGATIVEINTEGER (entier)
XSD_UNSIGNEDLONG (entier)
XSD_UNSIGNEDINT (entier)
XSD_UNSIGNEDSHORT (entier)
XSD_UNSIGNEDBYTE (entier)
XSD_POSITIVEINTEGER (entier)
XSD_NMTOKENS (entier)
XSD_ANYTYPE (entier)
SOAP_ENC_OBJECT (entier)
SOAP_ENC_ARRAY (entier)
XSD_1999_TIMEINSTANT (entier)
XSD_NAMESPACE (chaîne de caractères)
XSD_1999_NAMESPACE (chaîne de caractères)

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Classes pré-définies](#)
- [Constantes pré-définies](#)

- is_soap_fault
- SoapClient->__call()
- SoapClient->__construct()
- SoapClient->doRequest()
- SoapClient->getFunctions()
- SoapClient->getLastRequest()
- SoapClient->getLastRequestHeaders()
- SoapClient->getLastResponse()
- SoapClient->getLastResponseHeaders()
- SoapClient->getTypes()
- SoapClient->setCookie()
- SoapClient->soapCall()
- SoapFault->__construct()
- SoapHeader->__construct()
- SoapParam->__construct()
- SoapServer->addFunction()
- SoapServer->__construct()
- SoapServer->fault()
- SoapServer->getFunctions()
- SoapServer->handle()
- SoapServer->setClass()
- SoapServer->setPersistence()
- SoapVar->__construct()
- use_soap_error_handler

8.121.8 SoapClient->__call()() : Appelle une fonction SOAP (dépréciée)

SoapClient mixed **__call** (string *function_name* , array *arguments* , array *options* , array *input_headers* , array *output_headers*)

Cette méthode est dépréciée. Utilisez SoapClient->__soapCall() à la place.

8.121.9 SoapClient->__construct()() : Constructeur SoapClient

SoapClient **__construct** mixed *wSDL* array *options*

Ce constructeur crée des objets SoapClient dans le mode WSDL ou non-WSDL .

8.121.10 SoapClient->__doRequest()() : Effectue une requête SOAP

SoapClient string **__doRequest** (string *request* , string *location* , string *action* , int *version*)

Effectue une requête SOAP via HTTP.

Cette méthode peut être écrasée dans les sous-classes pour implémenter différents transporteurs, effectuer des opérations XML supplémentaires ou toute autre chose.

8.121.11 SoapClient->__getFunctions()() : Retourne une liste de fonctions SOAP

SoapClient array __getFunctions ()

Retourne une liste de fonctions SOAP.

Note
Cette fonction fonctionne uniquement en mode WSDL.

8.121.12 SoapClient->__getLastRequest()() : Retourne la dernière requête SOAP

SoapClient string __getLastRequest ()

Note
Cette méthode fonctionne uniquement si l'objet SoapClient a été créé avec l'option trace .

8.121.13 SoapClient->__getLastRequestHeaders()() : Retourne les en-têtes de la dernière requête SOAP

SoapClient string __getLastRequestHeaders ()

Note
Cette méthode fonctionne uniquement si l'objet SoapClient a été créé avec l'option trace .

8.121.14 SoapClient->__getLastResponse()() : Retourne la dernière réponse SOAP

SoapClient string __getLastResponse ()

Note
Cette méthode fonctionne uniquement si l'objet SoapClient a été créé avec l'option trace .

8.121.15 SoapClient->__getLastResponseHeaders()() : Retourne les en-têtes de la dernière réponse SOAP

SoapClient string __getLastResponseHeaders ()

Note
Cette méthode fonctionne uniquement si l'objet SoapClient a été créé avec l'option trace .

8.121.16 SoapClient->__getTypes()() : Retourne une liste des types SOAP

SoapClient array __getTypes ()

Cette fonction fonctionne uniquement en mode WSDL.

8.121.17 SoapClient->__setCookie()() : Définit le cookie qui sera envoyé avec la requête SOAP

SoapClient void **__setCookie** (string name , string value)

Définit un cookie à envoyer avec la requête SOAP.

Note

L'appel à cette méthode affectera tous les appels suivants aux méthodes SoapClient .
--

8.121.18 SoapClient->__soapCall()() : Appel une fonction SOAP

SoapClient mixed **__soapCall** (string function_name , array arguments , array options , mixed input_headers , array output_headers)

Ceci est une fonction bas niveau de l'API qui est utilisée pour faire des appels SOAP. Habituellement, en mode WSDL, vous pouvez appeler simplement les fonctions SOAP comme des méthodes SoapClient . Cette méthode est pratique en mode non-WSDL lorsque soapaction est inconnu, uri est différente de la valeur par défaut ou lors de l'envoi et/ou de la réception d'en-têtes SOAP.

En cas d'erreur, un appel à une fonction SOAP peut causer un lancement d'exception par PHP ou retourner un objet SoapFault si les exceptions sont désactivées. Pour vérifier si l'appel à cette fonction n'arrive pas à attraper les exceptions SoapFault, vérifiez le résultat avec la fonction [is_soap_fault](#) .

8.121.19 SoapFault->__construct()() : Constructeur SoapFault

SoapFault **__construct** string faultcode string faultstring string faultactor mixed detail string faultname SoapHeader headerfault

Cette classe est très utile lorsque vous voulez envoyer des réponses SOAP depuis le gestionnaire PHP. faultcode , faultstring , faultactor et details sont les éléments standards SOAP.

8.121.20 SoapHeader->__construct()() : Constructeur SoapHeader

SoapHeader **__construct** string namespace string name mixed data bool mustUnderstand mixed actor

Construit un nouvel objet SoapHeader .

8.121.21 SoapParam->__construct()() : Constructeur SoapParam

SoapParam **__construct** mixed data string name

Construit un nouvel objet SoapParam .

8.121.22 SoapServer->addFunction()() : Ajoute une ou plusieurs fonctions qui vont gérer les requêtes SOAP

SoapServer void **addFunction** (mixed functions)

Exporte une ou plusieurs fonctions pour les clients distants.

8.121.23 SoapServer->__construct()() : Constructeur SoapServer

SoapServer **__construct** mixed wsdl array options

Ce constructeur autorise la création des objets SoapServer en mode WSDL ou non-WSDL.

8.121.24 SoapServer->fault()() : Provoque une erreur SoapServer indiquant une erreur

SoapServer void **fault** (string code , string string , string actor , mixed details , string name)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.121.25 SoapServer->getFunctions()() : Retourne la liste des fonctions définies

SoapServer array **getFunctions** ()

Cette méthode retourne la liste de toutes les fonctions ajoutées par SoapServer->addFunction() ou SoapServer->setClass() .

8.121.26 SoapServer->handle()() : Gère une requête SOAP

SoapServer void **handle** (string soap_request)

Effectue une requête SOAP, appelle les fonctions nécessaire et envoie une réponse en retour.

8.121.27 SoapServer->setClass()() : Définit la classe qui gère les requêtes SOAP

SoapServer void **setClass** (string class_name , mixed args , mixed ...)

Exporte toutes les méthodes depuis une classe spécifiée.

Cet objet peut être rendu persistant pour une session donnée PHP en utilisant la méthode SoapServer->setPersistence() .

8.121.28 SoapServer->setPersistence()() : Définit le mode persistant de SoapServer

SoapServer void **setPersistence** (int mode)

Cette fonction permet de sauvegarder les données entre les requêtes dans une session PHP. Cela ne fonctionne qu'avec un serveur qui exporte les fonctions depuis une classe avec SoapServer->setClass() .

8.121.29 SoapVar->__construct()() : Constructeur SoapVar

SoapVar **__construct** mixed data int encoding string type_name string type_namespace string node_name string node_namespace

Construit un nouvel objet SoapVar .

8.121.30 use_soap_error_handler() : Utilisation ou non du gestionnaire d'erreurs SOAP et retourne l'ancienne valeur

bool **use_soap_error_handler** (bool handler)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.122 Sockets

8.122.1 Introduction

L'extension socket implémente une interface bas niveau avec les fonctions de communication par socket, basées sur les sockets BSD si populaires, et fournit la possibilité de fonctionner aussi bien sous forme de client que de serveur.

Pour une interface socket cliente plus générique, voyez [stream_socket_client](#) , [stream_socket_server](#) , [fsockopen](#) et [pfsockopen](#) .

Lorsque vous utilisez ces fonctions, il est important de vous rappeler que si de nombreuses fonctions ont le même nom que leur équivalent en langage C, elles ont souvent des déclarations différentes. Lisez attentivement les descriptions pour éviter des confusions.

Cela dit, ceux qui ne sont pas familiers avec la programmation par socket peuvent toujours trouver beaucoup de documentation dans les pages de manuel Unix appropriées, et il y a une grande quantité d'introductions en langage C sur le web, qui peuvent être facilement réutilisées, avec des adaptations mineures. [UNIX Socket FAQ](#) est un bon début.

8.122.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.122.3 Installation

Les fonctions de socket décrites ici font partie d'une extension PHP qui doit être activée lors de la compilation en utilisant l'option `--enable-sockets` , avec la commande `configure` .

Note
Le support des adresses IPv6 a été ajouté en PHP 5.0.

8.122.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.122.5 Types de ressources

Cette extension ne définit aucune ressource.

8.122.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

`AF_UNIX` ([entier](#))
`AF_INET` ([entier](#))

AF_INET6 ([entier](#))
 SOCK_STREAM ([entier](#))
 SOCK_DGRAM ([entier](#))
 SOCK_RAW ([entier](#))
 SOCK_SEQPACKET ([entier](#))
 SOCK_RDM ([entier](#))
 MSG_OOB ([entier](#))
 MSG_WAITALL ([entier](#))
 MSG_PEEK ([entier](#))
 MSG_DONTROUTE ([entier](#))
 SO_DEBUG ([entier](#))
 SO_REUSEADDR ([entier](#))
 SO_KEEPALIVE ([entier](#))
 SO_DONTROUTE ([entier](#))
 SO_LINGER ([entier](#))
 SO_BROADCAST ([entier](#))
 SO_OOBINLINE ([entier](#))
 SO_SNDBUF ([entier](#))
 SO_RCVBUF ([entier](#))
 SO_SNDLOWAT ([entier](#))
 SO_RCVLOWAT ([entier](#))
 SO_SNDTIMEO ([entier](#))
 SO_RCVTIMEO ([entier](#))
 SO_TYPE ([entier](#))
 SO_ERROR ([entier](#))
 SOL_SOCKET ([entier](#))
 PHP_NORMAL_READ ([entier](#))
 PHP_BINARY_READ ([entier](#))
 SOL_TCP ([entier](#))
 SOL_UDP ([entier](#))

8.122.7 Erreurs de socket

L'extension socket a été écrite pour fournir une interface utilisable avec les puissantes sockets fournies par BSD. Un soin particulier a été apporté pour que les fonctions soient aussi efficaces sous Unix que sous Windows. Presque toutes les fonctions de sockets peuvent échouer dans certaines circonstances, et émettent ainsi un message d'alerte E_WARNING décrivant l'erreur. Parfois, cela ne se fait pas selon les souhaits du développeur. Par exemple, la fonction `socket_read` peut tout à coup émettre un message E_WARNING car la connexion a été inopinément interrompue. Il est commun de supprimer les erreurs avec l'opérateur @ et de traiter les erreurs avec la fonction `socket_last_error`, au niveau de l'application. Vous pouvez appeler `socket_strerror` avec le code d'erreur pour connaître le message d'erreur, humainement lisible. Reportez-vous à leur description pour plus de détails.

Note

Les messages E_WARNING générés par l'extension socket sont en anglais, mais ils s'afficheront en fonction de la configuration locale (LC_MESSAGES) :

```
Warning - socket_bind() unable to bind address [98]: Die Adresse wird bereits verwendet
```

8.122.8 Exemples

Exemple de socket : serveur TCP/IP simple

Cet exemple montre comment créer un serveur simple. Changez les variables address et port pour vous adapter à votre configuration, et exécutez-le. Vous pourrez alors vous connecter au serveur avec une commande comme ceci : telnet 192.168.1.53 10000 (en supposant que l'adresse et le port correspondent à votre configuration). Tout ce que vous saisissez après cela sera envoyé au serveur, et affiché en retour. Pour vous déconnecter, tapez 'quit'.

```
#!/usr/local/bin/php -q
<?php
error_reporting(E_ALL);

/* Autorise l'exécution infinie du script, en attente de connexion. */
set_time_limit(0);

/* Active le vidage implicite des buffers de sortie, pour que nous
 * puissions voir ce que nous lisons au fur et à mesure. */
ob_implicit_flush();

$address = '192.168.1.53';
$port = 10000;

if (($sock = socket_create(AF_INET, SOCK_STREAM, SOL_TCP)) < 0) {
    echo "socket_create() a échoué : raison : " . socket_strerror($sock) . "\n";
}

if (($ret = socket_bind($sock, $address, $port)) < 0) {
    echo "socket_bind() a échoué : raison : " . socket_strerror ($ret) . "\n";
}

if (($ret = socket_listen($sock, 5)) < 0) {
    echo "socket_listen() a échoué : raison : " . socket_strerror ($ret) . "\n";
}

do {
    if (($msgsock = socket_accept($sock)) < 0) {
        echo "socket_accept() a échoué : raison : " . socket_strerror ($msgsock) . "\n";
        break;
    }
    /* Send instructions. */
    $msg = "\Bienvenue sur le serveur de test PHP.\n" .
        "Pour quitter, tapez 'quit'. Pour éteindre le serveur, tapez 'shutdown'.\n";
    socket_write($msgsock, $msg, strlen($msg));

    do {
        if (false === ($buf = socket_read($msgsock, 2048))) {
            echo "socket_read() a échoué : raison : " . socket_strerror ($ret) . "\n";
            break 2;
        }
        if (!$buf = trim($buf)) {
            continue;
        }
        if ($buf == 'quit') {
            break;
        }
        if ($buf == 'shutdown') {
            socket_close($msgsock);
            break 2;
        }
        $talkback = "PHP: You said '$buf'.\n";
        socket_write($msgsock, $talkback, strlen($talkback));
        echo "$buf\n";
    } while (true);
}
```

```

        socket_close($msgsock);
    } while (true);

socket_close($sock);
?>

```

Exemple avec les sockets : client TCP/IP simple

Cet exemple implémente un client HTTP simple. Il se connecte simplement à une page, envoie une requête HEAD , affiche le résultat et se termine.

```

<?php
error_reporting(E_ALL);

echo "<h2>Connexion TCP/IP</h2>\n";

/* Lit le port du service WWW. */
$service_port = getservbyname('www', 'tcp');

/* Lit l'adresse IP du serveur de destination */
$address = gethostbyname('www.example.com');

/* Cree une socket TCP/IP. */
$socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
if ($socket < 0) {
    echo "socket_create() a échoué : raison : " . socket_strerror($socket) . "\n";
} else {
    echo "OK.\n";
}

echo "Essai de connexion à '$address' sur le port '$service_port'...";
$result = socket_connect($socket, $address, $service_port);
if ($result < 0) {
    echo "socket_connect() a échoué : raison : ($result) " . socket_strerror($result) . "\n";
} else {
    echo "OK.\n";
}

$in = "HEAD / HTTP/1.0\r\n\r\n";
$in .= "Host: www.example.com\r\n";
$in .= "Connection: Close\r\n\r\n";
$out = '';

echo "Envoi de la requête HTTP HEAD...";
socket_write($socket, $in, strlen($in));
echo "OK.\n";

echo "Lire la réponse : \n\n";
while ($out = socket_read($socket, 2048)) {
    echo $out;
}

echo "Fermeture de la socket...";
socket_close($socket);
echo "OK.\n\n";
?>

```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)

- [Constantes pré-définies](#)
- [Erreurs de socket](#)
- [Exemples](#)
- [socket_accept](#)
- [socket_bind](#)
- [socket_clear_error](#)
- [socket_close](#)
- [socket_connect](#)
- [socket_create_listen](#)
- [socket_create_pair](#)
- [socket_create](#)
- [socket_get_option](#)
- [socket_getpeername](#)
- [socket_getsockname](#)
- [socket_last_error](#)
- [socket_listen](#)
- [socket_read](#)
- [socket_recv](#)
- [socket_recvfrom](#)
- [socket_select](#)
- [socket_send](#)
- [socket_sendto](#)
- [socket_set_block](#)
- [socket_set_nonblock](#)
- [socket_set_option](#)
- [socket_shutdown](#)
- [socket_strerror](#)
- [socket_write](#)

8.122.10 [socket_bind\(\)](#) : Lie un nom à une socket

bool [socket_bind](#) (resource socket , string address , int port)

Lie le nom donné par address à l'interface de connexion décrite par socket . Ceci doit être effectué avant qu'une connexion puisse être établie en utilisant [socket_connect](#) ou [socket_listen](#) .

8.122.11 [socket_clear_error\(\)](#) : Efface toutes les erreurs précédemment générées par une socket

void [socket_clear_error](#) (resource socket)

[socket_clear_error](#) efface tous les codes d'erreur qui ont été enregistrés pour la socket socket , ou bien pour la socket générale.

[socket_clear_error](#) permet de remettre à zéro les codes d'erreur d'une socket ou de la socket globale. Cela peut être utile pour détecter l'avènement d'une erreur durant une partie de l'application.

Voir aussi [socket_last_error](#) et [socket_strerror](#) .

8.122.12 `socket_close()` : Ferme une socket

void **socket_close** (resource socket)

`socket_close` ferme la socket `socket` et détruit toutes les ressources qui y sont liées.

Note

`socket_close` ne peut pas être utilisée sur les ressources de fichiers créées par `fopen` , `popen` , `fsockopen` , ou `pssockopen` ; cette fonction est destinée à traiter les sockets créées par les fonctions `socket_create` ou `socket_accept` .

Voir aussi `socket_bind` , `socket_listen` , `socket_create` et `socket_strerror` .

8.122.13 `socket_connect()` : Crée une connexion sur une socket

bool **socket_connect** (resource socket , string address , int port)

`socket_connect` crée une nouvelle connexion en utilisant la ressource `socket socket` , qui doit être une ressource de socket valide, créée par la fonction `socket_create` .

Le paramètre d'adresse `address` peut être soit une adresse IP numérique (i.e. 127.0.0.1) , si la socket est de la famille AF_INET ; ou bien le chemin d'une socket de domaine Unix, si la socket est de la famille AF_UNIX .

Le paramètre `port` n'est utilisé que lors de la connexion à une socket de type AF_INET , et désigne le port sur l'hôte distant, sur lequel une connexion doit être faite.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. Le code d'erreur généré peut être obtenu en appelant la fonction `socket_last_error` . Ce code d'erreur peut être passé à la fonction `socket_strerror` pour obtenir un message d'erreur humainement lisible.

Voir aussi `socket_bind` , `socket_listen` , `socket_create` , `socket_last_error` et `socket_strerror` .

8.122.14 `socket_create_listen()` : Ouvre une socket sur un port pour accepter les connexions

resource **socket_create_listen** (int port , int backlog)

`socket_create_listen` sert à simplifier la création de nouvelles sockets destinées à être mises en attente, et accepter de nouvelles connexions.

`socket_create_listen` crée une nouvelle ressource de socket, de type AF_INET , mise en attente sur **toutes** les interfaces locales, pour le port `port` .

Le paramètre `backlog` définit la taille maximum de la queue de connexions en attente. SOMAXCONN peut être utilisée comme valeur pour le paramètre `backlog` . Reportez-vous à `socket_listen` pour plus de détails.

`socket_create_listen` retourne une nouvelle ressource de socket en cas de succès et FALSE en cas d'erreur. Le code d'erreur généré peut être obtenu en appelant la fonction `socket_last_error` . Ce code d'erreur peut être passé à la fonction `socket_strerror` pour obtenir un message d'erreur humainement lisible.

Note
Si vous voulez créer une socket qui n'écoute que certaines interfaces, vous devez utiliser socket_create , socket_bind et socket_listen .
Voir aussi socket_create , socket_bind , socket_listen , socket_last_error et socket_strerror .

8.122.15 [socket_create_pair\(\)](#) : Crée une paire de sockets identiques et les stocke dans un tableau

bool [socket_create_pair](#) (int domain , int type , int protocol , array fd)

[socket_create_pair](#) crée une paire de sockets identiques et les stocke dans &fd . Cette fonction est utilisée couramment dans IPC (InterProcess Communication) .

Le paramètre domain spécifie la famille du protocole à utiliser par la socket.

Domaine	Description
AF_INET	Protocole IPv4. TCP et UDP sont les protocoles communs de cette famille. Supporté uniquement sous Windows.
AF_INET6	Protocole IPv6. TCP et UDP sont les protocoles communs de cette famille. Le support a été ajouté en PHP 5.0.0. Supporté uniquement sous Windows.
AF_UNIX	Protocole de communication locale. Le rendement élevé et des moindres coûts supplémentaires lui font une grande force d'IPC (Interprocess Communication) .

Le paramètre type spécifie le type de communication à utiliser par la socket.

Type	Description
SOCK_STREAM	Fournit des flux d'octets ordonnancés, fiables, full-duplex, raccordés sur la base. Un mécanisme de transmission des données " out-of-band " peut être supporté. Le protocole TCP est basé sur ce type de sockets.
SOCK_DGRAM	Support des datagrammes (moins de connexion, message non garanti d'une longueur maximum fixe). Le protocole UDP est basé sur ce type de sockets.
SOCK_SEQPACKET	Fournit un chemin de transmission de données séquentiel, fiable, connecté à la base par deux chemins pour les datagrammes de longueur maximale fixe ; un consommateur est requis pour lire la totalité d'un paquet avec chaque appel à la lecture.
SOCK_RAW	Fournit l'accès brut de protocole de réseau. Ce type spécial de sockets peut être utilisé pour construire manuellement tout type de protocole. Une utilisation commune de ce type de sockets est le traitement des requêtes ICMP (comme le ping, traceroute, etc.).
SOCK_RDM	Fournit une couche fiable de datagramme qui ne garantie pas l'ordre des données. Ce type de sockets est le plus susceptible de ne pas être implémenté sur votre système d'exploitation.

Le paramètre protocol définit un protocole spécifique dans le domaine spécifié domain pour être utilisé lors d'une communication sur une socket retournée. La valeur appropriée peut être retrouvée par son nom en utilisant la fonction [getprotobyname](#) . Si le protocole désiré est TCP ou UDP, les constantes correspondantes SOL_TCP et SOL_UDP peuvent être utilisées.

Nom	Description
icmp	Le protocole ICMP (Internet Control Message Protocol) est utilisé tout d'abord par les passerelles et les hôtes pour reporter les erreurs dans des communications de datagramme.

	La commande " ping " (présente dans les systèmes de production modernes) est un exemple d'application utilisant le protocole ICMP.
udp	Le protocole UDP (User Datagramm Protocol) est un protocole sans connexion, incertaine avec les longueurs d'enregistrement fixes. De ce fait, UDP requiert une quantité minimum de protocole.
tcp	Le protocole TCP (Transmission Control Protocol) est un protocole fiable, connecté sur la base, orienté flux et full-duplex. TCP garantit que chaque paquet est reçu dans l'ordre dans lequel il a été envoyé. Si quelques paquets sont perdus pendant la communication, TCP retransmettra ces paquets tant que l'hôte destinataire ne les aura pas reçu entièrement. Pour des raisons de fiabilité et de performance, l'implémentation TCP , elle-même, décide des frontières appropriées d'octets de la couche fondamentale de communication du datagramme. Par conséquent, les applications TCP doivent autoriser la possibilité de transmission partielle d'enregistrements.

Exemple avec `socket_create_pair`

```
<?php
$sockets = array();
/* Création de la paire de sockets */
if (!socket_create_pair(AF_UNIX, SOCK_STREAM, 0, $sockets)) {
    echo socket_strerror(socket_last_error());
}
/* Envoi et réception de données */
if (!socket_write($sockets[0], "ABCdef123\n", strlen("ABCdef123\n"))) {
    echo socket_strerror(socket_last_error());
}
if (!$data = socket_read($sockets[1], strlen("ABCdef123\n"), PHP_BINARY_READ)) {
    echo socket_strerror(socket_last_error());
}
var_dump($data);

/* Fermeture des sockets */
socket_close($sockets[0]);
socket_close($sockets[1]);
?>
```

Exemple IPC avec `socket_create_pair`

```
<?php
$array = array();
$strone = 'Message From Parent.';
$strtwo = 'Message From Child.';
if (!socket_create_pair(AF_UNIX, SOCK_STREAM, 0, $array)) {
    echo socket_strerror(socket_last_error());
}
$pid = pcntl_fork();
if ($pid == -1) {
    echo 'Impossible de dupliquer le processus.';
} elseif ($pid) {
    /* parent */
    socket_close($array[0]);
    if (!socket_write($array[1], $strone, strlen($strone))) {
        echo socket_strerror(socket_last_error());
    }
    if (socket_read($array[1], strlen($strtwo), PHP_BINARY_READ) == $strtwo) {
        echo "Réception de $strtwo\n";
    }
    socket_close($array[1]);
} else {
    /* fils */
    socket_close($array[1]);
    if (!socket_write($array[0], $strtwo, strlen($strtwo))) {
```

```

    echo socket_strerror(socket_last_error());
}
if (socket_read($ary[0], strlen($strone), PHP_BINARY_READ) == $strone) {
    echo "Réception de $strone\n";
}
socket_close($ary[0]);
}
?>

```

8.122.16 socket_create() : Crée une socket

resource **socket_create** (int domain , int type , int protocol)

socket_create crée un point de communication (une socket) et retourne une ressource de socket. Une connexion typique réseau est composée de deux sockets : une qui joue le rôle de client et l'autre celui du serveur.

Le paramètre domain spécifie la famille de protocoles à utiliser par la socket.

Domaine	Description
AF_INET	Protocole basé sur IPv4. TCP et UDP sont les protocoles communs de cette famille de protocoles.
AF_INET6	Protocole basé sur IPv6. TCP et UDP sont les protocoles communs de cette famille de protocoles. Le support a été ajouté en PHP 5.0.0.
AF_UNIX	Famille de protocoles locales de communication. Le rendement élevé et des moindres coûts supplémentaires lui font une grande force d'IPC (Interprocess Communication).

Le paramètre type sélectionne le type de communication à utiliser par la socket.

Type	Description
SOCK_STREAM	Fournit des flux d'octets ordonnancés, fiables, full-duplex, raccordés sur la base. Un mécanisme de transmission des données " out-of-band " peut être supporté. Le protocole TCP est basé sur ce type de sockets.
SOCK_DGRAM	Support des datagrammes (moins de connexion, message non garanti d'une longueur maximum fixe). Le protocole UDP est basé sur ce type de sockets.
SOCK_SEQPACKET	Fournit un chemin de transmission de données séquentiel, fiable, connecté à la base par deux chemins pour les datagrammes de longueur maximale fixe ; un consommateur est requis pour lire la totalité d'un paquet avec chaque appel à la lecture.
SOCK_RAW	Fournit l'accès brut de protocole de réseau. Ce type spécial de socket peut être utilisé pour construire manuellement tout type de protocole. Une utilisation commune de ce type de sockets est le traitement des requêtes ICMP (comme le ping, traceroute, etc.).
SOCK_RDM	Fournit une couche fiable de datagramme qui ne garantit pas l'ordre des données. Ce type de socket est le plus susceptible de ne pas être implémenté sur votre système d'exploitation.

Le paramètre protocol définit le protocole spécifique pour le domaine domain à utiliser lors de communications sur une socket retournée. La valeur appropriée peut être retrouvée par son nom en utilisant la fonction getprotobyname . Si le protocole désiré est TCP ou UDP, les constantes correspondantes SOL_TCP et SOL_UDP peuvent être utilisées.

Nom	Description
icmp	

	Le protocole ICMP (Internet Control Message Protocol) est utilisé tout d'abord par les passerelles et les hôtes pour reporter les erreurs dans des communications de datagramme. La commande " ping " (présente dans les systèmes de production modernes) est un exemple d'application utilisant le protocole ICMP.
udp	Le protocole UDP (User Datagramm Protocol) est un protocole sans connexion, incertain avec les longueurs d'enregistrements fixes. De ce fait, UDP requiert une quantité minimum de protocole aérienne.
tcp	Le protocole TCP (Transmission Control Protocol) est un protocole fiable, connecté sur la base, orienté flux et full-duplex. TCP garantit que chaque paquet est reçu dans l'ordre dans lequel il a été envoyé. Si quelques paquets sont perdus pendant la communication, TCP retransmettra ces paquets tant que l'hôte destinataire ne les aura pas reçus entièrement. Pour des raisons de fiabilité et de performance, l'implémentation TCP , elle-même, décide des frontières appropriées d'octets de la couche fondamentale de communication du datagramme. Par conséquent, les applications TCP doivent autoriser la possibilité de transmission partielle d'enregistrements.

socket_create retourne une ressource de socket en cas de succès et FALSE sinon. Le code d'erreur généré peut être obtenu en appelant la fonction socket_last_error . Ce code d'erreur peut être passé à la fonction socket_strerror pour obtenir un message d'erreur humainement lisible.

Note
Si une valeur invalide est spécifiée au paramètre domain ou au paramètre type , la fonction <u>socket_create</u> prendra comme paramètres par défaut respectivement AF_INET et SOCK_STREAM et générera un message d'alerte (E_WARNING).

Voir aussi socket_accept , socket_bind , socket_connect , socket_listen , socket_last_error et socket_strerror .

8.122.17 socket_get_option() : Lit les options de la socket

mixed socket_get_option (resource socket , int level , int optname)

socket_get_option retourne la valeur de l'option spécifiée par le paramètre optname pour la socket spécifiée par le paramètre socket . socket_get_option retourne FALSE en cas d'erreur.

Le paramètre level spécifie la couche de protocole de l'option. Par exemple, pour connaître les options de la couche socket, la valeur SOL_SOCKET du paramètre level sera utilisée. Les autres degrés, comme TCP , peuvent être utilisés en spécifiant le numéro du protocole de cette couche. Les numéros de protocoles peuvent être trouvés en utilisant la fonction getprotobyname .

Option	Description
SO_DEBUG	Reporte si les informations de débogage sont enregistrées ou pas.
SO_ACCEPTCONN	Reporte si l'écoute des sockets est activée ou pas.
SO_BROADCAST	Reporte si la transmission des annonces globales est supportée ou pas.
SO_REUSEADDR	Reporte si les adresses locales peuvent être réutilisées ou pas.
SO_KEEPALIVE	Reporte si les connexions sont persistantes avec des transmissions périodiques de messages ou pas. Si la socket connectée échoue en réponse à ces messages, la connexion est interrompue et le processus écrira à cette socket une notification avec un signal SIGPIPE.
SO_LINGER	Reporte si la socket socket s'attarde sur la fonction <u>socket_close</u> si des données sont présentes ou pas.
SO_OOBINLINE	Reporte si la socket socket part sur des données en ligne out-of-band ou pas.

SO_SNDBUF	Reporte les informations sur la taille du tampon envoyé.
SO_RCVBUF	Reporte les informations sur la taille du tampon reçu.
SO_ERROR	Reporte les informations sur le statut de l'erreur et le vide.
SO_TYPE	Reporte le type de la socket socket .
SO_DONTROUTE	Reporte si les messages sortants dévient les équipements standard de cheminement.
SO_RCVLOWAT	Reporte le nombre minimal d'octets au processus pour les opérations entrantes sur la socket socket . (Par défaut, 1)
SO_RCVTIMEO	Reporte la valeur du délai d'exécution pour les opérations entrantes.
SO_SNDLOWAT	Reporte le nombre minimal d'octets au processus pour les opérations sortantes sur la socket socket .
SO_SNDTIMEO	Reporte la valeur du délai d'exécution spécifiant le temps maximal d'exécution pour les fonctions sortantes bloquantes parce que la commande d'écoulement empêche des données d'être envoyé.
Note	
Cette fonction était appelée <code>socket_getopt()</code> avant PHP 4.3.0.	

8.122.18 `socket_getpeername()` : Interroge l'autre extrémité de la communication

`bool socket_getpeername (resource socket , string addr , int port)`

Si la socket socket est de type AF_INET , ou AF_INET6 `socket_getpeername` retournera l'**adresse IP** de l'hôte, en notation numérique (e.g. 127.0.0.1 ou fe80::1) dans le paramètre address , et si le paramètre optionnel port est présent, elle retournera aussi le port de la communication établie.

Si la socket socket est de type AF_UNIX , `socket_getpeername` retournera le chemin sur le système de fichiers (e.g. /var/run/daemon.sock) dans le paramètre address .

Note
<code>socket_getsockname</code> ne doit pas être utilisée avec les sockets AF_UNIX créées avec <code>socket_accept</code> . Seules les sockets créées avec <code>socket_connect</code> ou une socket serveur primaire suivant un appel à <code>socket_bind</code> retourneront des valeurs logiques.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. `socket_getpeername` peut aussi retourner FALSE si le type de la socket n'est ni AF_INET AF_INET6 , ni AF_UNIX , auquel cas le dernier code d'erreur socket n'est **pas** modifié.

Voir aussi `socket_getsockname` , `socket_last_error` et `socket_strerror` .

8.122.19 `socket_getsockname()` : Interroge la socket locale

`bool socket_getsockname (resource socket , string addr , int port)`

Si la socket socket est de type AF_INET , ou AF_INET6 , `socket_getsockname` retournera l'**adresse IP** locale, en notation numérique (e.g. 127.0.0.1 ou fe80::1) dans le paramètre address , et si le paramètre optionnel port est présent, elle retournera aussi le port de la communication établie.

Si la socket socket est de type AF_UNIX , `socket_getsockname` retournera le chemin sur le système de fichiers (e.g. /var/run/daemon.sock) dans le paramètre address .

Note

`socket_getsockname` ne doit pas être utilisée avec les sockets AF_UNIX créées avec `socket_connect`. Seules les sockets suivant un appel de `socket_bind` retourneront des valeurs logiques.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. `socket_getsockname` peut aussi retourner FALSE si le type de la socket n'est ni AF_INET, ni AF_INET6, ni AF_UNIX, auquel cas le dernier code d'erreur socket n'est **pas** modifié.

Voir aussi `socket_getpeername`, `socket_last_error` et `socket_strerror`.

8.122.20 `socket_last_error()` : Lit la dernière erreur générée par une socket

int `socket_last_error` (*resource socket*)

`socket_last_error` retourne un code d'erreur socket.

Si un paramètre est passé à cette fonction, la dernière erreur qui aura été générée par cette socket sera retournée. Si la ressource de socket est omise, le dernier code d'erreur généré est retourné. Ce comportement est particulièrement pratique pour des fonctions comme `socket_create` qui ne retourne pas de socket en cas d'échec, et `socket_select` qui peut échouer sans raison directement liée à la socket. Le code d'erreur peut être transmis à `socket_strerror` qui retourne un message d'erreur lisible.

```
<?php
    if (false == ($socket = @socket_create(AF_INET, SOCK_STREAM, SOL_TCP))) {
        die("Impossible de créer la socket, le code d'erreur est : " . socket_last_error() .
            ", le message d'erreur est : " . socket_strerror(socket_last_error()));
    }
?>
```

Note

`socket_last_error` n'efface pas le code d'erreur. Utilisez plutôt la fonction `socket_clear_error` pour cela.

8.122.21 `socket_listen()` : Attend une connexion sur une socket

bool `socket_listen` (*resource socket*, *int backlog*)

Une fois que la socket `socket` a été créée avec la fonction `socket_create` et liée à un nom avec la fonction `socket_bind`, elle peut être mise en attente de connexion entrante.

Un nombre maximum de backlog connexions seront mises en attente de traitement. Si une demande de connexion arrive et que la queue est pleine, le client recevra une erreur indiquant ECONNREFUSED, ou, si le protocole de support accepte les retransmissions, la requête sera ignorée pour que les tentatives ultérieures finissent par réussir.

Note

Le nombre maximum passé dans le paramètre backlog dépend essentiellement de la plate-forme de support. Sur Linux, il est tronqué automatiquement à SOMAXCONN. Sous Windows, si la constante SOMAXCONN est passée, le service responsable des sockets choisira une valeur maximum **raisonnable**. Il n'y a pas de méthode pour deviner la valeur réellement choisie.

socket_listen ne fonctionne qu'avec des sockets de type SOCK_STREAM et SOCK_SEQPACKET .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. Le code d'erreur généré peut être obtenu en appelant la fonction socket_last_error . Ce code d'erreur peut être passé à la fonction socket_strerror pour obtenir un message d'erreur humainement lisible.

Voir aussi socket_accept , socket_bind , socket_connect , socket_create et socket_strerror .

8.122.22 socket_read() : Lit des données d'une socket

string **socket_read** (resource socket , int length , int type)

socket_read lit des données dans la socket socket , qui doit être une ressource créée par socket_accept . Elle lit un maximum de length octets. Sinon, vous pouvez utiliser \r , \n ou \0 pour terminer la lecture (suivant la valeur choisie pour type , voir ci-dessous).

socket_read retourne les données sous forme de chaîne en cas de succès, et FALSE sinon (y compris si l'hôte distant a fermé la connexion). Le code d'erreur généré peut être obtenu en appelant la fonction socket_last_error . Ce code d'erreur peut être passé à la fonction socket_strerror pour obtenir un message d'erreur humainement lisible.

Note

<u>socket_read</u> retourne une chaîne de longueur zéro (""), lorsqu'il n'y a plus de données à lire.

Le paramètre optionnel type peut prendre l'une des valeurs constantes suivantes :

- PHP_BINARY_READ : utilise la fonction système recv() . Capable de lire les données binaires. (valeur par défaut en PHP >= 4.1.0)
- PHP_NORMAL_READ : la lecture s'arrête aux \n et \r (valeur par défaut en PHP <= 4.0.6).

Voir aussi socket_accept , socket_bind , socket_connect , socket_listen , socket_last_error , socket_strerror et socket_write .

8.122.23 socket_recv() : Reçoit des données d'une socket connectée

int **socket_recv** (resource socket , string buf , int len , int flags)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.122.24 socket_recvfrom() : Reçoit des données d'une socket, connectée ou pas

int **socket_recvfrom** (resource socket , string buf , int len , int flags , string name , int port)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

socket_recvfrom a été protégée des données binaires depuis PHP 4.3.0.

8.122.25 `socket_select()` : Exécute l'appel système `select()` un tableau de sockets avec une durée d'expiration

int `socket_select` (array read , array write , array except , int tv_sec , int tv_usec)

`socket_select` accepte un tableau de sockets et attend qu'elles changent de statut. Ceux qui sont familiers avec les sockets de BSD reconnaîtront dans ces tableaux de sockets les jeux de pointeurs de fichiers. Trois tableaux indépendants de ressources de sockets sont surveillés.

Les sockets listées dans le paramètre read seront surveillées en lecture : pour savoir quand elles sont disponibles en lecture (plus précisément, si une lecture ne va pas bloquer, en particulier, une ressource de socket a déjà atteint une fin de fichier, auquel cas `socket_read` retournera une chaîne de taille zéro).

Les sockets listées dans write seront surveillées en écriture : pour voir si une écriture ne va pas bloquer.

Les sockets listées dans except seront surveillées pour leurs exceptions.

<p>Attention</p> <p>En sortie de fonction, les tableaux sont modifiés pour indiquer quelles sockets ont changé d'état. Vous n'avez pas besoin de passer tous les tableaux à <code>socket_select</code> . Vous pouvez les omettre, ou utiliser un tableau vide, ou encore NULL à la place. N'oubliez pas que ces tableaux sont passés par référence et seront modifiés par <code>socket_select</code> .</p>
--

<p>Exemple avec <code>socket_select</code></p> <pre><?php /* Prépare le tableau read (socket surveillées en lecture) */ \$read = array(\$socket1, \$socket2); if (false === (\$num_changed_sockets = socket_select(\$read, \$write = NULL, \$except = NULL, 0))) { /* Gestion des erreurs */ } elseif (\$num_changed_sockets > 0) { /* Au moins une des sockets a été modifiée */ } ?></pre>

<p>Note</p> <p>A cause d'une limitation du moteur Zend actuel, il n'est pas possible de passer une constante comme NULL directement comme paramètre à cette fonction, qui attend une valeur par référence. A la place, utilisez un tableau temporaire ou une expression dont le membre de gauche est une variable temporaire :</p>

<p>Passer NULL à <code>socket_select</code></p> <pre><?php socket_select(\$r, \$w, \$e = NULL, 0); ?></pre>
--

Les paramètres tv_sec et tv_usec ensemble forment le paramètre timeout (durée de vie). Le timeout est la durée maximale de temps avant que `socket_select` ne se termine. tv_sec peut être zéro, ce qui fera que `socket_select` retournera immédiatement. C'est très pratique pour faire du polling (sondage). Si tv_sec est NULL (pas de timeout), `socket_select` peut se bloquer indéfiniment.

En cas de succès, `socket_select` retourne le nombre de sockets contenues dans les tableaux modifiés. Ce nombre peut être zéro si la durée maximale d'attente a été atteinte. En cas d'erreur,

FALSE est retourné. Le code d'erreur généré peut être obtenu en appelant la fonction `socket_last_error` . Ce code d'erreur peut être passé à la fonction `socket_strerror` pour obtenir un message d'erreur humainement lisible.

Note
Assurez-vous bien d'utiliser l'opérateur <code>===</code> lorsque vous vérifiez les erreurs. Etant donnée que <code>socket_select</code> peut retourner 0, la comparaison avec FALSE via <code>==</code> donnerait TRUE :
Analyser le résultat de <code>socket_select</code>
<pre><?php if (false === socket_select(\$r, \$w, \$e = NULL, 0)) { echo "socket_select() a échoué. Raison : " . socket_strerror(socket_last_error()) . "\n"; } ?></pre>

Note
Méfiez-vous des implémentations de sockets, qui doivent être manipulées avec délicatesse. Quelques règles de base :
<ul style="list-style-type: none"> • Vous devez toujours essayer d'utiliser <code>socket_select</code> sans timeout. Votre programme ne devrait avoir rien à faire si il n'y a pas de données disponibles. Le code qui dépend d'un timeout est généralement peu portable, et difficile à déboguer. • Une socket ne doit pas être ajoutée à l'un des tableaux en paramètre, si vous ne souhaitez pas vérifier le résultat après l'appel à <code>socket_select</code> . Après le retour de <code>socket_select</code> , toutes les sockets dans tous les tableaux doivent être vérifiées. Toute socket qui est disponible en écriture ou en lecture doit être utilisée pour écrire ou lire. • Si vous écrivez ou lisez avec une socket retournée dans un tableau, soyez conscient qu'elle ne pourra pas écrire ou lire toutes les données que vous demandez. Soyez prêt à ne pouvoir lire qu'un seul octet. • Il est commun à la plupart des implémentations de socket que la seule exception interceptée par les sockets dans le tableau except est le cas des données hors limites, reçues par une socket.

Voir aussi `socket_read` , `socket_write` , `socket_last_error` et `socket_strerror` .

8.122.26 `socket_send()` : Envoie des données à une socket connectée

int `socket_send` (resource socket , string buf , int len , int flags)

La fonction `socket_send` envoie len octets à la socket socket depuis le buffer buf .

Le paramètre flags peut être la combinaison des valeurs suivantes :

0x1	Traite les données OOB (out-of-band).
0x2	Traite le message entrant (peek).
0x4	Ignore le routage, utilise une interface directe.
0x8	Les données complètent l'enregistrement.
0x100	Les données complètent la transaction.

Voir aussi `socket_sendmsg` et `socket_sendto` .

8.122.27 `socket_sendto()` : Envoie une message à une socket, qu'elle soit connectée ou pas

int `socket_sendto` (resource socket , string buf , int len , int flags , string addr , int port)

`socket_sendto` envoie len octets issus du buffer buf via la socket socket , vers le port port , à l'adresse addr

Le paramètre flags peut prendre l'une des valeurs suivantes :

0x1	Traite les données OOB (out-of-band).
0x2	Traite le message entrant (peek).
0x4	Ignore le routage, utilise une interface directe.
0x8	Les données complètent l'enregistrement.
0x100	Les données complètent la transaction.

Exemple avec `socket_sendto`

```
<?php
$sh = socket_create(AF_INET,SOCK_STREAM,SOL_TCP);
if (socket_bind($sh, '127.0.0.1', 4242)) {
    echo "Socket connectée correctement";
}

$buf = 'Test Message';
$len = strlen($buf);

if (socket_sendto($sh, $buf, $len, 0x100, '192.168.0.2', 4242) !== FALSE) {
    echo "Message envoyé correctement";
}

socket_close($sh);
?>
```

Voir aussi `socket_send` et `socket_sendmsg` .

8.122.28 `socket_set_block()` : Met la socket en mode bloquant

bool `socket_set_block` (resource socket)

`socket_set_block` supprime l'option O_NONBLOCK de la socket spécifiée par socket .

Exemple avec `socket_set_block`

```
<?php
$port = 9090;
if (!$socket = socket_create_listen($port)) {
    echo socket_strerror(socket_last_error());
}

if (!socket_set_option($socket, SOL_SOCKET, SO_REUSEADDR, 1)) {
    echo socket_strerror(socket_last_error());
}

if (!socket_set_nonblock($socket)) { // $socket est maintenant non bloquante
    echo socket_strerror(socket_last_error());
}
```

```

if (!socket_set_block($socket)) { // $socket est maintenant bloquante
    echo socket_strerror(socket_last_error());
}
?>

```

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [socket_set_nonblock](#) et [socket_set_option](#)

8.122.29 socket_set_nonblock() : Sélectionne le mode non bloquant d'un descripteur de fichier

bool **socket_set_nonblock** (resource socket)

La fonction [socket_set_nonblock](#) configure l'option O_NONBLOCK pour la socket spécifiée par le paramètre socket .

Exemple avec [socket_set_nonblock](#)

```

<?php
$port = 9090;
if (!$socket = socket_create_listen($port)) {
    echo socket_strerror(socket_last_error());
}

if (!socket_set_option($socket, SOL_SOCKET, SO_REUSEADDR, 1)) {
    echo socket_strerror(socket_last_error());
}

if (!socket_set_nonblock($socket)) {
    echo socket_strerror(socket_last_error());
}
?>

```

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [socket_set_block](#) et [socket_set_option](#)

8.122.30 socket_set_option() : Modifie les options de socket

bool **socket_set_option** (resource socket , int level , int optname , mixed optval)

[socket_set_option](#) configure l'option spécifiée par optname , au niveau de protocole level à la valeur pointée par optval pour la socket spécifiée par socket . [socket_set_option](#) retourne FALSE en cas d'échec.

Le paramètre level spécifie la couche du protocole de l'option. Par exemple, pour modifier une option de la couche socket, un niveau égal à SOL_SOCKET va être utilisé. Les autres niveaux, comme TCP, peuvent être utilisés en spécifiant un numéro de protocole pour ce niveau. Les numéros de protocoles peuvent être utilisés en utilisant la fonction [getprotobyname](#) .

Les options disponibles sont les mêmes que pour [socket_get_option](#) .

Note

Cette fonction était appelée `socket_setopt` avant PHP 4.3.0.

8.122.31 `socket_shutdown()` : Eteint une socket en lecture et/ou écriture

bool `socket_shutdown` (resource `socket` , int `how`)

`socket_shutdown` vous permet d'empêcher les données entrantes ou sortantes ou les deux (par défaut) d'être émises via la socket `socket`

La valeur de `how` peut être l'une des suivantes :

0	Empêche la lecture de la socket
1	Empêche l'écriture de la socket
2	Empêche l'écriture et la lecture de la socket

8.122.32 `socket_strerror()` : Retourne une chaîne décrivant un message d'erreur

string `socket_strerror` (int `errno`)

`socket_strerror` prend un code d'erreur comme paramètre `errno` . Cette valeur est souvent retournée par la fonction `socket_last_error` . La fonction retourne le message d'erreur correspondant. Cela rend plus facile la compréhension des problèmes survenus. Par exemple, au lieu d'avoir à rechercher un fichier système dans lequel il y a l'explication de '-111', vous avez juste à le passer à `socket_strerror` , qui vous dira ce qui s'est passé.

Exemple avec `socket_strerror`

```
<?php
if (false == (@socket_create(AF_INET, SOCK_STREAM, SOL_TCP))) {
    echo "socket_create() a échoué : raison : " . socket_strerror(socket_last_error()) . "\n";
}

if (false == (@socket_bind($socket, '127.0.0.1', 80))) {
    echo "socket_bind() a échoué : raison : " . socket_strerror(socket_last_error($socket)) . "\n";
}
?>
```

L'affichage attendu pour l'exemple du dessus (en supposant que vous essayer d'exécuter le script sans les droits de root) :

```
socket_bind() a échoué : raison : Permission denied
```

Voir aussi `socket_accept` , `socket_bind` , `socket_connect` , `socket_listen` et `socket_create` .

8.122.33 `socket_write()` : Ecrit dans une socket

int `socket_write` (resource `socket` , string `buffer` , int `length`)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS

PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

socket_write écrit dans la socket socket les données du buffer buffer .

Le paramètre optionnel length peut spécifier explicitement la taille des données qui doivent être écrites. Si cette longueur est plus grande que la taille du buffer, elle sera ramenée automatiquement à la taille du buffer lui-même.

socket_write retourne le nombre d'octets qui ont pu être écrits dans la socket, ou bien FALSE en cas d'erreur. Le code d'erreur généré peut être obtenu en appelant la fonction socket_last_error . Ce code d'erreur peut être passé à la fonction socket_strerror pour obtenir un message d'erreur, humainement lisible.

Note

socket_write n'écrit pas nécessairement tous les octets du buffer fourni. Il est valide que, suivant certaines configuration de buffer réseau, seulement une certaine quantité de données, même un octet, soit écrit, y compris si votre buffer est plus grand. Vous devez alors vous assurer que vous n'avez pas oublié de transmettre le reste de vos données.

Note

Il est parfaitement valide pour socket_write de retourner zéro, ce qui signifie qu'aucun octet n'a été écrit. Soyez bien sûr d'utiliser l'opérateur === pour comparer le retour de la fonction avec FALSE , et détecter un cas d'erreur.

Voir aussi socket_accept , socket_bind , socket_connect , socket_listen , socket_read et socket_strerror .

8.123 Fonctions Standard PHP Library (SPL)

8.123.1 Introduction

SPL est une collection d'interfaces et de classes qui résolvent des problèmes standards.

Array

8.123.2 Installation

Cette extension est valable et compilée par défaut dans PHP 5.

8.123.3 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

RIT_LEAVES_ONLY ([entier](#))
RIT_SELF_FIRST ([entier](#))
RIT_CHILD_FIRST ([entier](#))
CIT_CALL_TOSTRING ([entier](#))
CIT_CATCH_GET_CHILD ([entier](#))

- [Introduction](#)
- [Installation](#)
- [Constantes pré-définies](#)
- [ArrayIterator::current](#)
- [ArrayIterator::key](#)
- [ArrayIterator::next](#)
- [ArrayIterator::rewind](#)
- [ArrayIterator::seek](#)
- [ArrayIterator::valid](#)
- [ArrayObject::append](#)
- [ArrayObject::__construct](#)
- [ArrayObject::count](#)
- [ArrayObject::getIterator](#)
- [ArrayObject::offsetExists](#)
- [ArrayObject::offsetGet](#)
- [ArrayObject::offsetSet](#)
- [ArrayObject::offsetUnset](#)
- [CachingIterator::hasNext](#)
- [CachingIterator::next](#)
- [CachingIterator::rewind](#)
- [CachingIterator::__toString](#)
- [CachingIterator::valid](#)
- [CachingRecursiveIterator::getChildren](#)
- [CachingRecursiveIterator::hasChildren](#)
- [class_implements](#)
- [class_parents](#)
- [DirectoryIterator::__construct](#)

- [DirectoryIterator::current](#)
- [DirectoryIterator::getATime](#)
- [DirectoryIterator::getChildren](#)
- [DirectoryIterator::getCTime](#)
- [DirectoryIterator::getFilename](#)
- [DirectoryIterator::getGroup](#)
- [DirectoryIterator::getInode](#)
- [DirectoryIterator::getMTime](#)
- [DirectoryIterator::getOwner](#)
- [DirectoryIterator::getPath](#)
- [DirectoryIterator::getPathname](#)
- [DirectoryIterator::getPerms](#)
- [DirectoryIterator::getSize](#)
- [DirectoryIterator::getType](#)
- [DirectoryIterator::isDir](#)
- [DirectoryIterator::isDot](#)
- [DirectoryIterator::isExecutable](#)
- [DirectoryIterator::isFile](#)
- [DirectoryIterator::isLink](#)
- [DirectoryIterator::isReadable](#)
- [DirectoryIterator::isWritable](#)
- [DirectoryIterator::key](#)
- [DirectoryIterator::next](#)
- [DirectoryIterator::rewind](#)
- [DirectoryIterator::valid](#)
- [FilterIterator::current](#)
- [FilterIterator::getInnerIterator](#)
- [FilterIterator::key](#)
- [FilterIterator::next](#)
- [FilterIterator::rewind](#)
- [FilterIterator::valid](#)
- [iterator_count](#)
- [iterator_to_array](#)
- [LimitIterator::getPosition](#)
- [LimitIterator::next](#)
- [LimitIterator::rewind](#)
- [LimitIterator::seek](#)
- [LimitIterator::valid](#)
- [ParentIterator::getChildren](#)
- [ParentIterator::hasChildren](#)
- [ParentIterator::next](#)
- [ParentIterator::rewind](#)
- [RecursiveDirectoryIterator::getChildren](#)
- [RecursiveDirectoryIterator::hasChildren](#)
- [RecursiveDirectoryIterator::key](#)
- [RecursiveDirectoryIterator::next](#)
- [RecursiveDirectoryIterator::rewind](#)
- [RecursiveIteratorIterator::current](#)
- [RecursiveIteratorIterator::getDepth](#)
- [RecursiveIteratorIterator::getSubIterator](#)
- [RecursiveIteratorIterator::key](#)
- [RecursiveIteratorIterator::next](#)
- [RecursiveIteratorIterator::rewind](#)
- [RecursiveIteratorIterator::valid](#)

- [SimpleXMLIterator::current](#)
- [SimpleXMLIterator::getChildren](#)
- [SimpleXMLIterator::hasChildren](#)
- [SimpleXMLIterator::key](#)
- [SimpleXMLIterator::next](#)
- [SimpleXMLIterator::rewind](#)
- [SimpleXMLIterator::valid](#)
- [spl_classes](#)

8.123.5 ArrayIterator::key() : Retourne la clé courante du tableau

mixed **ArrayIterator::key** (void)

[ArrayIterator::key](#) retourne la clé courante du tableau.

Exemple avec [ArrayIterator::key](#)

```
<?php
$array = array('key' => 'value');

$arrayobject = new ArrayObject($array);
$iterator = $arrayobject->getIterator();

echo $iterator->key(); //key
?>
```

8.123.6 ArrayIterator::next() : Se déplace vers la prochaine entrée

void **ArrayIterator::next** (void)

[ArrayIterator::next](#) déplace l'itérateur vers la prochaine entrée.

Exemple avec [ArrayIterator::next](#)

```
<?php
$arrayobject = new ArrayObject();

$arrayobject[] = 'zéro';
$arrayobject[] = 'un';

$iterator = $arrayobject->getIterator();

while($iterator->valid()) {
    echo $iterator->key() . ' => ' . $iterator->current() . "\n";

    $iterator->next();
}
?>
```

L'exemple ci-dessus va afficher :

```
0 => zéro
1 => un
```

8.123.7 ArrayIterator::rewind() : Revient à la position initiale

void **ArrayIterator::rewind** (void)

ArrayIterator::rewind remet l'itérateur au début du cycle.

Exemple avec ArrayIterator::rewind

```

<?php
$arrayobject = new ArrayObject();

$arrayobject[] = 'zero';
$arrayobject[] = 'un';
$arrayobject[] = 'deux';

$iterator = $arrayobject->getIterator();

$iterator->next();
echo $iterator->key(); //1

$iterator->rewind(); //retour au début
echo $iterator->key(); //0
?>

```

8.123.8 ArrayIterator::seek() : Avance à une position

void **ArrayIterator::seek** (int position)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.9 ArrayIterator::valid() : Vérifie si un tableau contient d'autres entrées

bool **ArrayIterator::valid** (void)

ArrayIterator::valid vérifie si un tableau contient d'autres entrées.

Exemple avec ArrayIterator::valid

```

<?php
$array = array('1' => 'un');

$arrayobject = new ArrayObject($array);
$iterator = $arrayobject->getIterator();

var_dump($iterator->valid()); //bool(true)

$iterator->next(); // élément suivant

//bool(false) car il n'y a qu'un seul élément
var_dump($iterator->valid());
?>

```

8.123.10 ArrayObject::append() : Ajoute la valeur à la fin

void **ArrayObject::append** (mixed newval)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.11 `ArrayObject::__construct()` : Construit un nouvel objet tableau

`ArrayObject` **`ArrayObject::__construct`** (mixed input)

`ArrayObject::__construct` construit un nouvel objet tableau. Le paramètre `input` accepte un tableau ou un autre objet `ArrayObject`.

Exemple avec <code>ArrayObject::__construct</code>
--

```
<?php
$array = array('1' => 'un',
              '2' => 'deux',
              '3' => 'trois');

$arrayobject = new ArrayObject($array);

var_dump($arrayobject);
?>
```

L'exemple ci-dessus va afficher :

```
object(ArrayObject)#1 (3) {
  [1]=>
  string(3) "un"
  [2]=>
  string(3) "deux"
  [3]=>
  string(5) "trois"
}
```

8.123.12 `ArrayObject::count()` : Retourne le nombre d'éléments dans l'itérateur

int **`ArrayObject::count`** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.13 `ArrayObject::getIterator()` : Créé un nouvel itérateur à partir d'une instance `ArrayObject`

`ArrayIterator` **`ArrayObject::getIterator`** (void)

`ArrayObject::getIterator` retourne un itérateur à partir d'un objet de classe `ArrayObject`.

Exemple avec <code>ArrayObject::getIterator</code>
--

```
<?php
$array = array('1' => 'un',
              '2' => 'deux',
              '3' => 'trois');
```

```
$arrayobject = new ArrayObject($array);  
$iterator = $arrayobject->getIterator();  
while($iterator->valid()) {  
    echo $iterator->key() . ' => ' . $iterator->current() . "\n";  
    $iterator->next();  
}  
?>
```

L'exemple ci-dessus va afficher :

```
1 => un  
2 => deux  
3 => trois
```

8.123.14 ArrayObject::offsetExists() : Retourne si \$index existe

bool **ArrayObject::offsetExists** (mixed index)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.15 ArrayObject::offsetGet() : Retourne la valeur à l'\$index spécifié

bool **ArrayObject::offsetGet** (mixed index)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.16 ArrayObject::offsetSet() : Définit \$newval comme valeur à l'\$index spécifié

void **ArrayObject::offsetSet** (mixed index , mixed newval)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.17 ArrayObject::offsetUnset() : Efface la valeur à l'\$index spécifié

void **ArrayObject::offsetUnset** (mixed index)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.18 CachingIterator::hasNext() : Vérifie si l'itérateur interne a un élément suivant valide

bool **CachingIterator::hasNext** (void)

8.123.13 ArrayObject::getIterator() : Créé un nouvel itérateur à partir d'une instance ArrayObject 803

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.19 CachingIterator::next() : Déplace l'itérateur à la position suivante

void **CachingIterator::next** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.20 CachingIterator::rewind() : Remplace l'itérateur au début

void **CachingIterator::rewind** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.21 CachingIterator::__toString() : Retourne la représentation de l'élément courant sous forme de chaîne

string **CachingIterator::__toString** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.22 CachingIterator::valid() : Vérifie si l'élément courant est valide

bool **CachingIterator::valid** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.23 CachingRecursiveIterator::getChildren() : Retourne le fils de l'itérateur interne comme un CachingRecursiveIterator

CachingRecursiveIterator **CachingRecursiveIterator::getChildren** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.24 CachingRecursiveIterator::hasChildren() : Vérifie si l'élément courant de l'itérateur interne a un fils

boolean **CachingRecursiveIterator::hasChildren** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.25 `class_implements()` : Retourne les interfaces implémentées dans une classe donnée

array `class_implements` (mixed `class` , *bool* `autoload`)

Cette fonction retourne un tableau avec les noms des interfaces que la classe `class` ainsi que ses parents implémentent.

8.123.26 `class_parents()` : Retourne la classe parente d'une classe

array `class_parents` (mixed `class` , *bool* `autoload`)

`class_parents` retourne un tableau avec le nom des classes parentes de la classe `class` .

8.123.27 `DirectoryIterator::__construct()` : Construit un nouvel itérateur de dossier à partir d'un chemin

`DirectoryIterator` `DirectoryIterator::__construct` (string `path`)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.28 `DirectoryIterator::current()` : Retourne this (requis pour l'interface `Iterator`)

`DirectoryIterator` `DirectoryIterator::current` (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.29 `DirectoryIterator::getATime()` : Récupère la date et l'heure du dernier accès à un fichier

int `DirectoryIterator::getATime` (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.30 `DirectoryIterator::getChildren()` : Retourne un itérateur pour l'entrée courante si c'est un répertoire

`RecursiveDirectoryIterator` `DirectoryIterator::getChildren` (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.31 DirectoryIterator::getCTime() : Récupère l'heure de modification de l'inode d'un fichier

int DirectoryIterator::getCTime (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.32 DirectoryIterator::getFilename() : Retourne le nom de l'entrée courante du dossier

string DirectoryIterator::getFilename (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.33 DirectoryIterator::getGroup() : Récupère le groupe d'un fichier

int DirectoryIterator::getGroup (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.34 DirectoryIterator::getInode() : Récupère l'inode d'un fichier

int DirectoryIterator::getInode (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.35 DirectoryIterator::getMTime() : Récupère l'heure de la dernière modification d'un fichier

int DirectoryIterator::getMTime (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.36 DirectoryIterator::getOwner() : Récupère le propriétaire d'un fichier

int DirectoryIterator::getOwner (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.37 DirectoryIterator::getPath() : Retourne le chemin du dossier

string **DirectoryIterator::getPath** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.38 DirectoryIterator::getPathname() : Retourne le chemin et le nom de l'entrée courante du dossier

string **DirectoryIterator::getPathname** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.39 DirectoryIterator::getPerms() : Récupère les permissions d'un fichier

int **DirectoryIterator::getPerms** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.40 DirectoryIterator::getSize() : Récupère la taille d'un fichier

int **DirectoryIterator::getSize** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.41 DirectoryIterator::getType() : Récupère le type d'un fichier

string **DirectoryIterator::getType** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.42 DirectoryIterator::isDir() : Retourne TRUE si le fichier est un dossier

bool **DirectoryIterator::isDir** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.43 DirectoryIterator::isDot() : Retourne TRUE si l'entrée courante est '.' ou '..'

bool **DirectoryIterator::isDot** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.44 DirectoryIterator::isExecutable() : Retourne TRUE si le fichier est exécutable

bool **DirectoryIterator::isExecutable** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.45 DirectoryIterator::isFile() : Retourne TRUE si l'entrée est un fichier valide

bool **DirectoryIterator::isFile** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.46 DirectoryIterator::isLink() : Retourne TRUE si le fichier est un lien symbolique

bool **DirectoryIterator::isLink** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.47 DirectoryIterator::isReadable() : Retourne TRUE si le fichier est accessible en lecture

bool **DirectoryIterator::isReadable** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.48 DirectoryIterator::isWritable() : Retourne TRUE si le fichier peut être modifié

bool **DirectoryIterator::isWritable** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.49 DirectoryIterator::key() : Retourne l'entrée courante du dossier

string **DirectoryIterator::key** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.50 DirectoryIterator::next() : Se déplace vers la prochaine entrée

void **DirectoryIterator::next** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.51 DirectoryIterator::rewind() : Revient au début du dossier

void **DirectoryIterator::rewind** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.52 DirectoryIterator::valid() : Vérifie si le répertoire contient encore des entrées

string **DirectoryIterator::valid** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.53 FilterIterator::current() : Récupère la valeur de l'élément courant

mixed **FilterIterator::current** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.54 FilterIterator::getInnerIterator() : Récupère l'itérateur interne

Iterator **FilterIterator::getInnerIterator** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.55 FilterIterator::key() : Récupère la clé courante

mixed **FilterIterator::key** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.56 FilterIterator::next() : Déplace l'itérateur à la position suivante

void **FilterIterator::next** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.57 FilterIterator::rewind() : Remplace l'itérateur au début

void **FilterIterator::rewind** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.58 FilterIterator::valid() : Vérifie si l'élément courant est valide

bool **FilterIterator::valid** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.59 iterator_count() : Compte de nombre d'éléments dans un itérateur

int **iterator_count** (IteratorAggregate iterator)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.60 iterator_to_array() : Copie un itérateur dans un tableau

array **iterator_to_array** (IteratorAggregate iterator)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.61 LimitIterator::getPosition() : Retourne la position courante

int **LimitIterator::getPosition** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.62 LimitIterator::next() : Déplace l'itérateur à la position suivante

void **LimitIterator::next** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.63 LimitIterator::rewind() : Remplace l'itérateur au début

void **LimitIterator::rewind** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.64 LimitIterator::seek() : Place l'itérateur à une position donnée

void **LimitIterator::seek** (int position)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.65 LimitIterator::valid() : Vérifie si l'élément courant est valide

bool **LimitIterator::valid** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.66 ParentIterator::getChildren() : Retourne le fils de l'itérateur interne contenu dans ParentIterator

ParentIterator **ParentIterator::getChildren** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.67 ParentIterator::hasChildren() : Vérifie si l'élément courant de l'itérateur interne a un fils

bool **ParentIterator::hasChildren** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.68 ParentIterator::next() : Déplace l'itérateur à la position suivante

void **ParentIterator::next** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.69 ParentIterator::rewind() : Remplace l'itérateur au début

void **ParentIterator::rewind** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.70 RecursiveDirectoryIterator::getChildren() : Retourne un itérateur pour l'entrée courante si c'est un dossier

object RecursiveDirectoryIterator::getChildren (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.71 RecursiveDirectoryIterator::hasChildren() : Vérifie si l'entrée courante est un dossier et n'est pas '.' ou '..'

bool RecursiveDirectoryIterator::hasChildren (bool allow_links)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.72 RecursiveDirectoryIterator::key() : Retourne le chemin et le nom de l'entrée courante du dossier

string RecursiveDirectoryIterator::key (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.73 RecursiveDirectoryIterator::next() : Se déplace vers la prochaine entrée

void RecursiveDirectoryIterator::next (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.74 RecursiveDirectoryIterator::rewind() : Revient à la position initiale dans le dossier

void RecursiveDirectoryIterator::rewind (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.75 RecursiveIteratorIterator::current() : Accède à la valeur de l'élément courant

mixed RecursiveIteratorIterator::current (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.76 **RecursiveIterator::getDepth() : Récupère la profondeur courante de la récursivité de l'itérateur**

int **RecursiveIterator::getDepth** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.77 **RecursiveIterator::getSubIterator() : L'itérateur secondaire actif courant**

RecursiveIterator **RecursiveIterator::getSubIterator** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.78 **RecursiveIterator::key() : Accède à la clé courante**

mixed **RecursiveIterator::key** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.79 **RecursiveIterator::next() : Déplace l'itérateur à la position suivante**

void **RecursiveIterator::next** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.80 **RecursiveIterator::rewind() : Replace l'itérateur au début**

void **RecursiveIterator::rewind** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.81 **RecursiveIterator::valid() : Vérifie si la position courante est valide**

boolean **RecursiveIterator::valid** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.82 SimpleXMLIterator::current() : Retourne l'entrée courante de SimpleXML

mixed SimpleXMLIterator::current (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.83 SimpleXMLIterator::getChildren() : Retourne un itérateur pour l'entrée courante, si c'est un objet SimpleXML

object SimpleXMLIterator::getChildren (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.84 SimpleXMLIterator::hasChildren() : Indique si l'entrée courante de SimpleXML est un objet

bool SimpleXMLIterator::hasChildren (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.85 SimpleXMLIterator::key() : Retourne la clé courante SimpleXML

mixed SimpleXMLIterator::key (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.86 SimpleXMLIterator::next() : Va à l'entrée SimpleXML suivante

void SimpleXMLIterator::next (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.87 SimpleXMLIterator::rewind() : Replace le pointeur SimpleXML au début

void SimpleXMLIterator::rewind (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.88 SimpleXMLIterator::valid() : Vérifie si une ressource SimpleXML contient d'autres entrées

bool **SimpleXMLIterator::valid** (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.123.89 spl_classes() : Retourne les classes SPL disponibles

array **spl_classes** (void)

spl_classes retourne un tableau avec la liste des classes SPL actuellement disponibles.

Exemple avec spl_classes

```
<?php
print_r(spl_classes());
?>
```

L'exemple ci-dessus va afficher quelque chose de similaire à :

```
Array
(
  [ArrayObject] => ArrayObject
  [ArrayIterator] => ArrayIterator
  [CachingIterator] => CachingIterator
  [CachingRecursiveIterator] => CachingRecursiveIterator
  [DirectoryIterator] => DirectoryIterator
  [FilterIterator] => FilterIterator
  [LimitIterator] => LimitIterator
  [ParentIterator] => ParentIterator
  [RecursiveDirectoryIterator] => RecursiveDirectoryIterator
  [RecursiveIterator] => RecursiveIterator
  [RecursiveIteratorIterator] => RecursiveIteratorIterator
  [SeekableIterator] => SeekableIterator
  [SimpleXMLIterator] => SimpleXMLIterator
)
```

8.124 Fonctions SQLite

8.124.1 Introduction

Cette extension fait l'interface avec le moteur de base SQLite Embeddable SQL Database Engine . SQLite est une bibliothèque C qui implémente un moteur de bases de données SQL embarqué. Les programmes qui utilisent SQLite peuvent disposer d'un accès à une base de données SQL, sans avoir à faire tourner d'autre processus serveur.

SQLite n'est pas un logiciel client pour se connecter à un grand nom des bases de données. SQLite est le serveur. La bibliothèque SQLite lit et écrit directement dans les fichiers de bases de données sur le disque.

Note

Pour plus d'informations sur SQLite , voyez le site web http://sqlite.org/ .

8.124.2 Installation

Lisez le fichier d'installation, qui est fourni avec la distribution. Ou bien, utilisez simplement l'installateur PEAR : `pear install sqlite` . SQLite est déjà fourni avec PHP et vous n'avez pas besoin de télécharger d'autres logiciels.

Les utilisateurs de Windows peuvent télécharger la version DLL de l'extension SQLite ici : [php_sqlite.dll](#) .

Dans la version 5 de PHP, l'extension SQLite ainsi que le moteur sont intégrés et compilés par défaut. Cependant, depuis PHP 5.1.0, vous devez activer manuellement cette extension dans le `php.ini` (parcequ'il est maintenant inclus en tant que partagé) en ajoutant la ligne suivante : `php_extension=php_sqlite.dll` .

Note

Installation sur Windows pour les comptes sans privilèges

Sur les systèmes d'exploitation Windows, les comptes sans privilèges n'ont pas la variable d'environnement TMP d'initialisée par défaut. Cela fera créer des fichiers temporaires par sqlite dans le dossier de windows, ce qui n'est pas voulu. Alors, vous devriez initialiser la variable d'environnement TMP pour le serveur web ou dans le compte utilisateur où le serveur web fonctionne. Si Apache est votre serveur web, vous pouvez accomplir cela via une directive SetEnv dans votre fichier <code>httpd.conf</code> . Par exemple :
--

<code>SetEnv TMP c:/temp</code>

Si vous êtes incapable d'établir cette configuration à ce niveau du serveur, vous pouvez implémenter cette configuration dans votre script :
--

<code>putenv('TMP=C:/temp');</code>

La configuration doit référer à un répertoire à l'intérieur duquel le serveur web a la permission de créer des fichiers et bien sûr d'y écrire et supprimer les fichiers créés. Autrement, vous recevrez le message d'erreur suivant :
--

malformed database schema -

unable to open a temporary database file for storing temporary tables

8.124.3 Pré-requis

Afin de disposer de ces fonctions, vous devez compiler PHP avec le support SQLite , ou charger dynamiquement l'extension SQLite depuis votre fichier php.ini .

8.124.4 Types de ressources

Il y a deux ressources, utilisées par l'extension SQLite . La première est une connexion à la base et la seconde est un résultat de requête.

8.124.5 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Les fonctions `sqlite_fetch_array` et `sqlite_current` utilisent des constantes pour spécifier les différents types de résultats. Les constantes sont les suivantes :

Constantes de résultats SQLite

`SQLITE_ASSOC` (entier)

Les colonnes sont retournées dans le tableau, en utilisant le nom de la colonne comme nom d'index.

`SQLITE_BOTH` (entier)

Les colonnes sont retournées dans le tableau, en utilisant simultanément le nom de la colonne comme nom d'index et un index numérique.

`SQLITE_NUM` (entier)

Les colonnes sont retournées dans le tableau, en utilisant une indexation numérique. L'index commence à 0, pour identifier le premier champ.

Un certain nombre de fonctions peuvent retourner des codes d'état. Les constantes suivantes sont définies :

Constantes codes d'état SQLite

`SQLITE_OK` (entier)

Résultat réussi.

`SQLITE_ERROR` (entier)

Erreur SQL ou base de données manquante.

`SQLITE_INTERNAL` (entier)

Une erreur interne de logique dans SQLite.

`SQLITE_PERM` (entier)

Permission d'accès refusée.

`SQLITE_ABORT` (entier)

Routine de procédure de rappel a demandé un abandon.

`SQLITE_BUSY` (entier)

Le fichier de base de données est verrouillé.

`SQLITE_LOCKED` (entier)

Une table dans la base de données est verrouillée.

`SQLITE_NOMEM` (entier)

Allocation de mémoire échouée.

`SQLITE_READONLY` (entier)

- Essai d'écrire dans une base de données en lecture seule.
- SQLITE_INTERRUPT (entier)
Opération terminée de manière interne.
- SQLITE_IOERR (entier)
Erreur disque I/O s'est produite.
- SQLITE_CORRUPT (entier)
L'image disque de la base de données est malformée.
- SQLITE_NOTFOUND (entier)
(Interne) Table ou enregistrement non trouvé.
- SQLITE_FULL (entier)
Insertion échouée car la base de données est pleine.
- SQLITE_CANTOPEN (entier)
Impossible d'ouvrir le fichier de base de données.
- SQLITE_PROTOCOL (entier)
Erreur du protocole de verrou de base de données.
- SQLITE_EMPTY (entier)
(Interne) Une table de la base de données est vide.
- SQLITE_SCHEMA (entier)
Le schéma de base de données a changé.
- SQLITE_TOOBIG (entier)
Trop de données pour une ligne de la table.
- SQLITE_CONSTRAINT (entier)
Arrêt dû à une violation de contrainte.
- SQLITE_MISMATCH (entier)
Type de données incorrect.
- SQLITE_MISUSE (entier)
Bibliothèque utilisée incorrectement.
- SQLITE_NOLFS (entier)
Utilisation de fonctionnalités de l'OS non supportées sur l'hôte.
- SQLITE_AUTH (entier)
Autorisation échouée.
- SQLITE_ROW (entier)
Processus interne a une autre ligne prête.
- SQLITE_DONE (entier)
Processus interne a terminé l'exécution.

8.124.6 Classes pré-définies

8.124.6.1 SQLiteDatabase

Représente une base de données SQLite ouverte.

8.124.6.1.1 Constructeur

- __construct - construit un nouvel objet SQLiteDatabase

8.124.6.1.2 Méthodes

- query - Exécute une requête
- queryExec - Exécute une requête de résultat
- arrayQuery - Exécute une requête et retourne les résultats dans un tableau

8.124.5 Constantes pré-définies

- singleQuery - Exécute une requête et retourne soit un tableau pour une seule colonne, soit la valeur de la première ligne
- unbufferedQuery - Exécute une requête non bufférisée
- lastInsertRowid - Retourne l'identifiant de la dernière ligne insérée
- changes - Retourne le nombre de lignes modifiées par la dernière requête
- createAggregate - Enregistre une UDF agrégeante pour les requêtes SQLite
- createFunction - Enregistre une fonction utilisateur "classique" UDF pour SQLite
- busyTimeout - Configure ou désactive le délai d'attente d'une base SQLite occupée
- lastErrr - Retourne le dernier code erreur de la dernière erreur rencontrée
- fetchColumnTypes - Retourne un tableau de types de colonnes depuis une table particulière

8.124.6.2 SQLiteResult

Représente un jeu de résultats SQLite bufférisé.

8.124.6.2.1 Méthodes

- fetch - Récupère la ligne suivante depuis un jeu de résultats en tant que tableau
- fetchObject - Récupère la ligne suivante depuis un jeu de résultats en tant qu'objet
- fetchSingle - Récupère la première ligne depuis le jeu de résultats en tant que chaîne de caractères
- fetchAll - Récupère toutes les lignes depuis le jeu de résultats en tant que tableau de tableaux
- column - Récupère une colonne depuis la ligne courante du jeu de résultats
- numFields - Retourne le numéro du champ depuis un jeu de résultats
- fieldName - Retourne le nom d'un champ particulier depuis le jeu de résultats
- current - Récupère la ligne courante depuis le jeu de résultat en tant que tableau
- key - Retourne l'index de la ligne courante
- next - Déplace le pointeur sur le numéro de ligne suivante
- valid - Retourne si oui ou non il reste des lignes de disponibles
- rewind - Déplace le pointeur sur le numéro de la première ligne d'un jeu de résultats
- prev - Déplace le pointeur sur le numéro de ligne précédent du jeu de résultats
- hasPrev - Retourne si oui ou non une ligne précédente est disponible
- numRows - Retourne le nombre de lignes dans le jeu de résultats
- seek - Déplace le pointeur vers un numéro de ligne précis

8.124.6.3 SQLiteUnbuffered

Représente un jeu de résultats SQLite non-bufférisé. Les jeux de résultats non-bufférisés sont séquentiels, déplacement du pointeur uniquement vers le numéro suivant.

8.124.6.3.1 Méthodes

- fetch - Récupère la ligne suivante depuis un jeu de résultats en tant que tableau
- fetchObject - Récupère la ligne suivante depuis un jeu de résultats en tant qu'objet
- fetchSingle - Récupère la première colonne depuis un jeu de résultats en tant que chaîne de caractères
- fetchAll - Récupère toutes les lignes depuis le jeu de résultats en tant que tableau de tableaux
- column - Récupère une colonne depuis la ligne courante d'un jeu de résultats
- numFields - Retourne le nombre de champs dans le jeu de résultats

- fieldName - Retourne le nom d'un champ en particulier depuis le jeu de résultats
- current - Récupère la ligne courante depuis le jeu de résultats en tant que tableau
- next - Déplace le pointeur vers un numéro de ligne suivant
- valid - Retourne si oui ou non il reste des lignes de disponibles

8.124.7 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

Nom	Par défaut	Modifiable	Historique
sqlite.assoc_case	"0"	PHP_INI_ALL	Disponible depuis PHP 5.0.0.

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [Directives du php.ini](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

sqlite.assoc_case int

Utilise des index à casse libre (0), majuscule (1) ou minuscule (2).

Cette option sert principalement lorsque vous avez besoin de la compatibilité avec d'autres systèmes de bases de données, où les noms des colonnes sont toujours retournés en majuscule ou en minuscule, indépendamment de la casse réelle de la colonne dans la base.

La bibliothèque SQLite retourne le nom de la colonne sans y toucher (c'est la casse qui sera utilisée dans votre base). Lorsque sqlite.assoc_case vaut 0 cette casse sera conservée. Lorsque cette option vaut 1 ou 2 , PHP va modifier la casse des noms, et mettre le nom de la colonne respectivement en majuscules ou minuscules.

Utiliser cette option implique une légère perte de performances, mais c'est BEAUCOUP plus rapide que de faire la gestion des caractères vous-même.

- [Introduction](#)
- [Installation](#)
- [Pré-requis](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Classes pré-définies](#)
- [Configuration à l'exécution](#)
- [sqlite_array_query](#)
- [sqlite_busy_timeout](#)
- [sqlite_changes](#)
- [sqlite_close](#)
- [sqlite_column](#)
- [sqlite_create_aggregate](#)
- [sqlite_create_function](#)
- [sqlite_current](#)
- [sqlite_error_string](#)
- [sqlite_escape_string](#)
- [sqlite_exec](#)
- [sqlite_factory](#)
- [sqlite_fetch_all](#)
- [sqlite_fetch_array](#)
- [sqlite_fetch_column_types](#)

- [sqlite_fetch_object](#)
- [sqlite_fetch_single](#)
- [sqlite_fetch_string](#)
- [sqlite_field_name](#)
- [sqlite_has_more](#)
- [sqlite_has_prev](#)
- [sqlite_key](#)
- [sqlite_last_error](#)
- [sqlite_last_insert_rowid](#)
- [sqlite_libencoding](#)
- [sqlite_libversion](#)
- [sqlite_next](#)
- [sqlite_num_fields](#)
- [sqlite_num_rows](#)
- [sqlite_open](#)
- [sqlite_popen](#)
- [sqlite_prev](#)
- [sqlite_query](#)
- [sqlite_rewind](#)
- [sqlite_seek](#)
- [sqlite_single_query](#)
- [sqlite_udf_decode_binary](#)
- [sqlite_udf_encode_binary](#)
- [sqlite_unbuffered_query](#)
- [sqlite_valid](#)

8.124.9 [sqlite_busy_timeout\(\)](#) : SQLiteDatabase->busyTimeout()

void **sqlite_busy_timeout** (resource dbhandle , int milliseconds)

Style orienté objet (méthode)

SQLiteDatabase void **busyTimeout** (int milliseconds)

Spécifie la durée maximale, en millisecondes, que SQLite attendra pour qu'une base de données dbhandle soit utilisable.

8.124.10 [sqlite_changes\(\)](#) : SQLiteDatabase->changes()

int **sqlite_changes** (resource dbhandle)

Style orienté objet (méthode)

SQLiteDatabase int **changes** (void)

[sqlite_changes](#) retourne le nombre de lignes qui ont été modifiées par la dernière requête SQLite, exécutée dans la base db .

8.124.11 `sqlite_close()` : Ferme la connexion à SQLite

void **sqlite_close** (resource dbhandle)

`sqlite_close` referme la connexion à la base SQLite, identifiée par la ressource dbhandle . Si la connexion à la base était persistante, elle sera fermée et supprimée de la liste des connexions persistantes.

8.124.12 `sqlite_column()` : `SQLiteResult->column()`

mixed **sqlite_column** (resource result , mixed index_or_name , *bool decode_binary*)
SQLiteResult mixed **column** (mixed index_or_name , *bool decode_binary*) SQLiteUnbuffered
mixed **column** (mixed index_or_name , *bool decode_binary*)

`sqlite_column` lit la valeur de la colonne nommée index_or_name (si c'est une chaîne), ou d'index index_or_name (si c'est un entier), dans le résultat SQLite result .

8.124.13 `sqlite_create_aggregate()` : `SQLiteDatabase->createAggregate()`

void **sqlite_create_aggregate** (resource dbhandle , string function_name , callback step_func , callback finalize_func , *int num_args*)

Style orienté objet (méthode)

SQLiteDatabase void **createAggregate** (string function_name , callback step_func , callback finalize_func , *int num_args*)

`sqlite_create_aggregate` est similaire à `sqlite_create_function` , car elle enregistre une fonction qui sera utilisée pour calculer un résultat agrégé sur plusieurs lignes d'une requête.

La différence principale entre cette fonction et `sqlite_create_function` est que deux fonctions sont nécessaires pour gérer les agrégations : `step_func` est appelée pour chaque ligne du résultat. Votre fonction PHP doit accumuler le résultat, et le stocker dans le contexte d'agrégation. Une fois que toutes les lignes ont été appelées, `finalize_func` sera appelée, et elle doit alors lire les données dans ce contexte d'agrégation, et en retourner le résultat. Les fonctions de callback doivent retourner un type compréhensible par SQLite (i.e. type scalaire).

8.124.14 `sqlite_create_function()` : `SQLiteDatabase->createFunction()`

void **sqlite_create_function** (resource dbhandle , string function_name , callback callback , *int num_args*)

Style orienté objet (méthode)

SQLiteDatabase void **createFunction** (string function_name , callback callback , *int num_args*)

`sqlite_create_function` vous permet d'enregistrer une fonction PHP dans SQLite comme UDF (Fonction Définie par l'Utilisateur, User Defined Function), pour qu'elle soit accessible depuis les requêtes SQL.

Les UDF peuvent être utilisées dans n'importe quelle requête SQL qui peut appeler des fonctions, telles que SELECT et UPDATE, ou même des triggers.

8.124.15 `sqlite_current()` : `SQLiteResult->current()`

array `sqlite_current` (resource `result` , int `result_type` , bool `decode_binary`)

Style orienté objet (méthode)

SQLiteResult array `current` (int `result_type` , bool `decode_binary`) SQLiteUnbuffered array `current` (int `result_type` , bool `decode_binary`)

`sqlite_current` est identique à `sqlite_fetch_array` hormis le fait qu'elle ne fait pas progresser le pointeur de résultat avant de lire des données. Elle ne fait que retourner les données qui sont disponibles à la position courante.

8.124.16 `sqlite_error_string()` : Retourne le message d'erreur SQLite

string `sqlite_error_string` (int `error_code`)

`sqlite_error_string` retourne un message d'erreur lisible, pour le numéro d'erreur `error_code` , retourné par `sqlite_last_error` .

8.124.17 `sqlite_escape_string()` : Protège une chaîne de caractères pour utilisation avec SQLite

string `sqlite_escape_string` (string `item`)

`sqlite_escape_string` va ajouter des guillemets dans la chaîne `item` , pour qu'elle puisse être utilisée correctement dans une requête SQL. Cela inclut notamment le doublement des guillemets simples ('), et la vérification des caractères binaires non sécuritaires, de la chaîne de requête.

Si `item` contient le caractère NUL et qu'il commence avec un caractère dont la valeur ordinaire est 0x01 , PHP va appliquer un schéma de codage, pour que vous puissiez stocker puis relire correctement ces données.

Bien que ce codage sécurise l'insertion des données, il va rendre la recherche de texte par simple comparaison, ou en utilisant la clause LIKE , inutilisable dans vos requêtes pour les colonnes qui contiennent ces données binaires. En pratique, cela ne devrait pas être un problème, car votre utilisation de la base devrait faire que vous n'utiliserez pas ces colonnes (en fait, il est mieux de stocker des données binaires dans d'autres systèmes, comme des fichiers).

Attention
<code>addslashes</code> NE doit PAS être utilisée pour protéger vos requêtes dans SQLite. Cela va conduire à d'étranges résultats lors de la lecture de vos données.
Note
N'utilisez pas cette fonction pour coder les valeurs retournées par des fonctions utilisateurs, créées avec les fonctions <code>sqlite_create_function</code> ou <code>sqlite_create_aggregate</code> : utilisez plutôt <code>sqlite_udf_encode_binary</code> .

8.124.18 sqlite_exec() : SQLiteDatabase->exec()

bool **sqlite_exec** (resource dbhandle , string query , string error_msg) bool **sqlite_exec** (string query , resource dbhandle)

Style orienté objet (méthode)

SQLiteDatabase bool **queryExec** (string query , string error_msg)

sqlite_exec exécute une requête SQL query sur une base de données spécifiée par la ressource dbhandle .

Attention

SQLite **doit** exécuter de multiples requêtes séparées par des points-virgules ; vous pouvez donc l'utiliser pour exécuter un script SQL que vous avez chargé depuis un fichier ou que vous avez intégré dans un script.

8.124.19 sqlite_factory() : Ouvre une base SQLite et crée un objet pour elle

SQLiteDatabase **sqlite_factory** (string filename , int mode , string error_message)

sqlite_factory fonctionne de la même façon que sqlite_open dans le fait qu'il ouvre une base de données SQLite ou tente de la créer si elle n'existe pas. Cependant, un objet SQLiteDatabase est retourné plutôt qu'une ressource. Lisez la page de référence sur sqlite_open pour son utilisation.

8.124.20 sqlite_fetch_all() : SQLiteResult->fetchAll()

array **sqlite_fetch_all** (resource result , int result_type , bool decode_binary)

Style orienté objet (méthode)

SQLiteResult array **fetchAll** (int result_type , bool decode_binary) SQLiteUnbuffered array **fetchAll** (int result_type , bool decode_binary)

Retourne un tableau des lignes restantes dans le jeu de résultats. Si appelée juste après sqlite_query , elle retourne toutes les lignes. Si appelée après sqlite_fetch_array , elle retourne le reste. S'il n'y a plus de lignes disponibles dans le jeu de résultats, elle retourne un tableau vide.

8.124.21 sqlite_fetch_array() : SQLiteResult->fetch()

array **sqlite_fetch_array** (resource result , int result_type , bool decode_binary)

Style orienté objet (méthode)

SQLiteResult array **fetch** (int result_type , bool decode_binary) SQLiteUnbuffered array **fetch** (int result_type , bool decode_binary)

sqlite_fetch_array lit la prochaine ligne dans le résultat result . S'il n'y a pas d'autres lignes, sqlite_fetch_array retourne FALSE et, sinon, elle retourne un tableau représentant les données lues.

8.124.22

sqlite_fetch_column_types() : SQLiteDatabase->fetchColumnTypes()

array **sqlite_fetch_column_types** (string *table_name* , resource *dbhandle* , int *result_type*)

Style orienté objet (méthode)

SQLiteDatabase array **fetchColumnTypes** (string *table_name* , int *result_type*)

sqlite_fetch_column_types retourne un tableau de types de colonnes depuis la table *table_name* spécifiée.

8.124.23 sqlite_fetch_object() : SQLiteResult->fetchObject()

object **sqlite_fetch_object** (resource *result* , string *class_name* , array *ctor_params* , bool *decode_binary*)

Style orienté objet (méthode)

SQLiteResult object **fetchObject** (string *class_name* , array *ctor_params* , bool *decode_binary*)
SQLiteUnbuffered object **fetchObject** (string *class_name* , array *ctor_params* , bool *decode_binary*)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.124.24 sqlite_fetch_single() : SQLiteResult->fetchSingle()

string **sqlite_fetch_single** (resource *result* , bool *decode_binary*)

Style orienté objet (méthode)

SQLiteResult string **fetchSingle** (bool *decode_binary*)
SQLiteUnbuffered string **fetchSingle** (bool *decode_binary*)

sqlite_fetch_single est identique à sqlite_fetch_array hormis le fait qu'elle retourne uniquement la valeur de la première colonne du résultat.

C'est la méthode optimale pour lire des données, si vous n'êtes intéressé que par les valeurs de la première colonne.

8.124.25 sqlite_fetch_string() : Alias de sqlite_fetch_single

Cette fonction est un alias de : sqlite_fetch_single .

8.124.26 sqlite_field_name() : SQLiteResult->fieldName()

string **sqlite_field_name** (resource *result* , int *field_index*)

Style orienté objet (méthode)

SQLiteResult string **fieldName** (int field_index) SQLiteUnbuffered string **fieldName** (int field_index)

sqlite_field_name retourne le nom du champ identifié par son index numérique field_index , dans le résultat result .

8.124.27 **sqlite_has_more()** : Indique s'il reste des lignes SQLite à lire

bool **sqlite_has_more** (resource result)

Recherche si d'autres lignes sont disponibles dans le jeu de résultats donné.

8.124.28 **sqlite_has_prev()** : SQLiteResult->hasPrev()

bool **sqlite_has_prev** (resource result)

Style orienté objet (méthode)

SQLiteResult bool **hasPrev** (void)

Recherche si d'autres lignes précédentes sont disponibles dans le jeu de résultats donné.

8.124.29 **sqlite_key()** : SQLiteResult->key()

int **sqlite_key** (resource result)

Style orienté objet (méthode)

SQLiteResult int **key** (void)

sqlite_key retourne l'index de la ligne courante d'un jeu de résultats bufférisé result .

8.124.30 **sqlite_last_error()** : SQLiteDatabase->lastError()

int **sqlite_last_error** (resource dbhandle)

Style orienté objet (méthode)

SQLiteDatabase int **lastError** (void)

sqlite_last_error retourne le code d'erreur de la dernière opération SQL effectuée sur la base dbhandle . Une description lisible de cette erreur peut être obtenue avec la fonction sqlite_error_string .

8.124.31 `sqlite_last_insert_rowid()` : `SQLiteDatabase->lastInsertRowid()`

int `sqlite_last_insert_rowid` (resource dbhandle)

Style orienté objet (méthode)

SQLiteDatabase int `lastInsertRowid` (void)

`sqlite_last_insert_rowid` retourne le numéro de ligne inséré dans la base dbhandle , si elle a été créée avec un champ auto-incrémenté.

Array

8.124.32 `sqlite_libencoding()` : Retourne l'encodage utilisé par la bibliothèque SQLite

string `sqlite_libencoding` (void)

La librairie SQLite peut être compilée avec le support de ISO-8859-1 ou UTF-8 . `sqlite_libencoding` vous permet de savoir quel encodage est utilisé.

Attention

La distribution par défaut de PHP compile libsqlite avec l'encodage ISO-8859-1 . Cependant, c'est un leurre. Au lieu d'utiliser ISO-8859-1 , elle va utiliser votre configuration locale pour les comparaisons et les tris. Il vaut donc mieux penser que ce n'est pas ISO-8859-1 , mais plutôt '8-bit'.

Lorsqu'elle est compilée avec le support UTF-8 , SQLite gère le codage et l'encodage des séquences multi-octets UTF-8 , mais ne fait pas totalement le travail lorsqu'elle opère avec les données (aucune normalisation n'est faite, par exemple), et certaines comparaisons peuvent être erronées.

Attention

Il n'est pas recommandé d'utiliser PHP en configuration de serveur web avec une version de SQLite qui utilise l'encodage UTF-8 , car libsqlite va interrompre le processus si elle détecte un problème avec l'encodage UTF-8 .

8.124.33 `sqlite_libversion()` : Retourne la version de la bibliothèque SQLite

string `sqlite_libversion` (void)

`sqlite_libversion` retourne la version de la bibliothèque SQLite.

8.124.34 `sqlite_next()` : `SQLiteResult->next()`

bool `sqlite_next` (resource result)

Style orienté objet (méthode)

SQLiteResult bool `next` (void) SQLiteUnbuffered bool `next` (void)

[sqlite_next](#) avance le pointeur de résultat à la prochaine ligne du résultat result .

8.124.35 [sqlite_num_fields\(\)](#) : SQLiteResult->numFields()

int [sqlite_num_fields](#) (resource result)

Style orienté objet (méthode)

SQLiteResult int **numFields** (void) SQLiteUnbuffered int **numFields** (void)

Retourne le nombre de champs dans le résultat SQLite result .

8.124.36 [sqlite_num_rows\(\)](#) : SQLiteResult->numRows()

int [sqlite_num_rows](#) (resource result)

Style orienté objet (méthode)

SQLiteResult int **numRows** (void)

[sqlite_num_rows](#) retourne le nombre de lignes dans le résultat SQLite result .

8.124.37 [sqlite_open\(\)](#) : Ouvre une base SQLite et la crée si elle n'existe pas

resource [sqlite_open](#) (string filename , int mode , string error_message)

Style orienté objet (constructeur) :

SQLiteDatabase **__construct** string filename int mode string error_message

Ouvre une base de données SQLite ou crée la base de données si elle n'existe pas.

8.124.38 [sqlite_popen\(\)](#) : Ouvre une connexion SQLite persistante et crée la base si elle n'existe pas

resource [sqlite_popen](#) (string filename , int mode , string error_message)

[sqlite_popen](#) fonctionne exactement comme [sqlite_open](#) , hormis le fait qu'elle utilise le mécanisme de ressources persistantes de PHP. Pour des détails sur la signification des paramètres, voyez la fonction [sqlite_open](#) .

[sqlite_popen](#) va d'abord vérifier qu'il reste une connexion persistante déjà ouverte pour le fichier filename . Si elle peut en trouver une, elle l'utilisera et, sinon, elle ouvrira une nouvelle connexion.

L'intérêt de cette méthode est que vous évitez de relire la base, les schémas de tables et d'index, pour chaque page exécutée par un serveur web SAPI persistant (tous les SAPI, sauf les CGI et CLI).

Note

Si vous utilisez une connexion persistante et que la base de données est modifiée par un autre processus (via une table de cron, par exemple), et que le processus recrée la base en l'écrasant, soit par effacement et reconstruction, ou par déplacement d'une nouvelle version à la place de l'ancienne, vous pourriez obtenir des comportements inattendus lorsque vous utiliserez une vieille connexion persistante sur une nouvelle base.

Pour éviter cette situation, faites en sorte que vos processus parallèles ouvrent les bases et fassent leurs modifications dans une transaction.

8.124.39 `sqlite_prev()` : `SQLiteResult->prev()`

bool **sqlite_prev** (resource result)

Style orienté objet (méthode)

SQLiteResult bool **prev** (void)

sqlite_prev se positionne sur la ligne précédente du jeu de résultats result .

8.124.40 `sqlite_query()` : `SQLiteDatabase->query()`

resource **sqlite_query** (resource dbhandle , string query , int result_type , string error_msg)
 resource **sqlite_query** (string query , resource dbhandle , int result_type , string error_msg)

Style orienté objet (méthode)

SQLiteDatabase SQLiteResult **query** (string query , int result_type , string error_msg)

Exécute la requête SQL query dans une base SQLite.

8.124.41 `sqlite_rewind()` : `SQLiteResult->rewind()`

bool **sqlite_rewind** (resource result)

Style orienté objet (méthode)

SQLiteResult bool **rewind** (void)

sqlite_rewind replace le pointeur à la première ligne du jeu de résultats donné.

8.124.42 `sqlite_seek()` : `SQLiteResult->seek()`

bool **sqlite_seek** (resource result , int rownum)

Style orienté objet (méthode)

SQLiteResult bool **seek** (int rownum)

sqlite_seek déplace le pointeur de résultat à la ligne rownum .

8.124.38 `sqlite_popen()` : Ouvre une connexion SQLite persistante et crée la base si elle n'existe pas

8.124.43 `sqlite_single_query()` : `SQLiteDatabase->singleQuery()`

array **`sqlite_single_query`** (resource `db` , string `query` , bool `first_row_only` , bool `decode_binary`)

Style orienté objet (méthode)

SQLiteDatabase array **`singleQuery`** (string `query` , bool `first_row_only` , bool `decode_binary`)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.124.44 `sqlite_udf_decode_binary()` : Décode des données binaires, passées à une UDF SQLite

string **`sqlite_udf_decode_binary`** (string `data`)

`sqlite_udf_decode_binary` décode des données binaires, qui auraient été encodées avec `sqlite_udf_encode_binary` ou `sqlite_escape_string` .

Vous devez appeler cette fonction sur les paramètres passés à votre UDF, si vous devez manipuler des données binaires, car le codage binaire de PHP va masquer le contenu original de la donnée.

PHP ne fait pas cette opération de codage/décodage automatiquement, car cela réduirait considérablement les performances.

8.124.45 `sqlite_udf_encode_binary()` : Encode les données binaires d'une UDF SQLite avant de les retourner

string **`sqlite_udf_encode_binary`** (string `data`)

`sqlite_udf_encode_binary` applique le codage aux données binaires `data` pour qu'elles puissent être correctement retournées depuis la requête (car la bibliothèque libsqlite n'est pas compatible avec les données binaires).

S'il y a des chances que vos données ne soient pas compatibles, (par exemple, si elles contiennent le caractère NUL au milieu de la chaîne, et non pas seulement à la fin, ou si le premier caractère est un caractère 0x01) alors vous devrez appeler cette fonction pour protéger vos données.

PHP ne fait pas cette opération de codage/décodage automatiquement, pour des raisons de performances.

Note
N'utilisez pas la fonction <code>sqlite_escape_string</code> pour protéger les données dans vos UDF, car cela conduira à doubler la protection. Utilisez plutôt cette fonction à la place.

8.124.46 `sqlite_unbuffered_query()` : `SQLiteDatabase->unbufferedQuery()`

resource **`sqlite_unbuffered_query`** (resource `dbhandle` , string `query` , int `result_type` , string `error_msg`) resource **`sqlite_unbuffered_query`** (string `query` , resource `dbhandle` , int

result_type , *string error_msg*)

Style orienté objet (méthode)

SQLiteDatabase SQLiteUnbuffered **unbufferedQuery** (*string query* , *int result_type* , *string error_msg*)

sqlite_unbuffered_query est identique à sqlite_query hormis le fait que le résultat retourné est une séquence croissante, qui ne peut être lue que dans un seul sens.

Cette fonction est idéale pour générer des tables HTML, où vous n'avez besoin que d'une ligne en même temps, et que vous n'avez pas à faire d'accès aléatoire aux lignes.

Note

Les fonctions telles que <u>sqlite_seek</u> et <u>sqlite_rewind</u> ne fonctionnent pas avec les résultats non bufférisés de cette fonction.
--

8.124.47 **sqlite_valid() : SQLiteResult->valid()**

bool **sqlite_valid** (*resource result*)

Style orienté objet (méthode)

SQLiteResult bool **valid** (*void*) SQLiteUnbuffered bool **valid** (*void*)

Recherche si d'autres lignes sont disponibles à partir de la ressource de résultat donnée.

8.125 Shell2 sécurisé

8.125.1 Introduction

La liaison à la bibliothèque `libssh2` fournit un accès à des ressources (shell, exécution distante, tunneling, transfert de fichiers) sur une machine distante en utilisant un transport crypté sécurisé.

8.125.2 Installation

Les binaires **Windows** peuvent être trouvés sur <http://snaps.php.net/> . Pour l'installation, téléchargez `php_ssh2.dll` dans le dossier spécifié par votre directive `extension_dir` du fichier `php.ini`. Activez-le en ajoutant `extension=php_ssh2.dll` dans votre fichier `php.ini` et redémarrez votre serveur web.

```
extension_dir=c:/php5/exts/
extension=php_ssh2.dll
```

Linux, BSD, et autres variantes *nix peuvent compiler cette extension en suivant les étapes suivantes :

- Téléchargez et installez [OpenSSL](#) . Si vous installez OpenSSL via le système de paquetages de votre distribution, assurez-vous d'installer également les bibliothèques de développement. Elles seront typiquement paquetées sous le nom `openssl-dev` , `openssl-devel` , ou quelque chose comme cela.
- Téléchargez et installez [libssh2](#) . Typiquement, cela signifie d'exécuter les commandes suivantes depuis les fichiers sources `libssh2` : `./configure && make all install` .
- Exécutez l'installeur Pear depuis PECL/ssh2: `pear install ssh2`
- Copiez le fichier `ssh2.so` depuis le dossier indiqué par le processus de compilation dans le dossier spécifié à l'intérieur du fichier spécifié par votre fichier `php.ini`, `extension_dir` .
- Ajoutez `extension=ssh2.so` dans votre `php.ini`.
- Redémarrez votre serveur web pour recharger votre configuration `php.ini`.

Note
Versions de développement
Il n'y a actuellement aucune version stable de PECL/ssh2, pour forcer l'installation d'une version beta de PECL/ssh2, exécutez : <code>pear install ssh2- beta</code>
Array

Des informations comme les notes sur les nouvelles versions, les téléchargements, les sources des fichiers, les informations concernant les mainteneurs ainsi qu'un CHANGELOG, peuvent être trouvées ici : <http://pecl.php.net/package/ssh2> .

Note
Vous avez besoin de la version 0.4 ou supérieure de la bibliothèque <code>libssh2</code> (ou supérieure, lisez les notes de versions).

8.125.3 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

- SSH2_FINGERPRINT_MD5 (integer)
Flag permettant à la fonction [ssh2_fingerprint](#) de demander l'empreinte de la clé de l'hôte en tant que hash MD5.
- SSH2_FINGERPRINT_SHA1 (integer)
Flag permettant à la fonction [ssh2_fingerprint](#) de demander l'empreinte de la clé de l'hôte en tant que hash SHA1.
- SSH2_FINGERPRINT_HEX (integer)
Flag permettant à la fonction [ssh2_fingerprint](#) de demander l'empreinte de la clé de l'hôte en tant que chaîne hexits.
- SSH2_FINGERPRINT_RAW (integer)
Flag permettant à la fonction [ssh2_fingerprint](#) de demander l'empreinte de la clé de l'hôte en tant que chaîne de caractères 8-bit.
- SSH2_TERM_UNIT_CHARS (integer)
Flag spécifiant à la fonction [ssh2_shell](#) que les paramètres width et height sont fournis sous la forme de taille de caractères.
- SSH2_TERM_UNIT_PIXELS (integer)
Flag spécifiant à la fonction [ssh2_shell](#) que les paramètres width et height sont fournis sous la forme de pixels.
- SSH2_DEFAULT_TERM_WIDTH (integer)
Largeur par défaut du terminal demandé par la fonction [ssh2_shell](#) .
- SSH2_DEFAULT_TERM_HEIGHT (integer)
Hauteur par défaut du terminal demandé par la fonction [ssh2_shell](#) .
- SSH2_DEFAULT_TERM_UNIT (integer)
Unité par défaut du terminal demandé par la fonction [ssh2_shell](#) .
- SSH2_STREAM_STDIO (integer)
Flag pour que la fonction [ssh2_fetch_stream](#) demande un sous-canal STDIO.
- SSH2_STREAM_STDERR (integer)
Flag pour que la fonction [ssh2_fetch_stream](#) demande un sous-canal STDERR.
- SSH2_DEFAULT_TERMINAL (string)
Type par défaut du terminal (e.g. vt102, ansi, xterm, vanilla) demandé par la fonction [ssh2_shell](#) .

- [Introduction](#)
- [Installation](#)
- [Constantes pré-définies](#)
- [ssh2_auth_hostbased_file](#)
- [ssh2_auth_none](#)
- [ssh2_auth_password](#)
- [ssh2_auth_pubkey_file](#)
- [ssh2_connect](#)
- [ssh2_exec](#)
- [ssh2_fetch_stream](#)
- [ssh2_fingerprint](#)
- [ssh2_methods_negotiated](#)
- [ssh2_publickey_add](#)
- [ssh2_publickey_init](#)
- [ssh2_publickey_list](#)
- [ssh2_publickey_remove](#)
- [ssh2_scp_rcv](#)

- [ssh2_scp_send](#)
- [ssh2_sftp_lstat](#)
- [ssh2_sftp_mkdir](#)
- [ssh2_sftp_readlink](#)
- [ssh2_sftp_realpath](#)
- [ssh2_sftp_rename](#)
- [ssh2_sftp_rmdir](#)
- [ssh2_sftp_stat](#)
- [ssh2_sftp_symlink](#)
- [ssh2_sftp_unlink](#)
- [ssh2_sftp](#)
- [ssh2_shell](#)
- [ssh2_tunnel](#)

8.125.5 `ssh2_auth_none()` : Identification en tant que "none"

mixed `ssh2_auth_none` (resource session , string username)

`ssh2_auth_none` tente une identification en tant que "none", qui, habituellement, échoue (et doit échouer). La particularité de cette fonctionnalité est la suivante : le serveur doit retourner un tableau de méthodes d'identification autorisées. Si le serveur accepte l'identification "none" en tant que méthode d'identification pour l'utilisateur username , cette fonction retournera TRUE .

Utilisation de `ssh2_auth_none` pour récupérer la liste des méthodes d'identification.

```
<?php
$connection = ssh2_connect('shell.example.com', 22);

$auth_methods = ssh2_auth_none($connection, 'user');

if (in_array('password', $auth_methods)) {
    echo "Le serveur supporte l'identification basée sur les mots de passe\n";
}
?>
```

8.125.6 `ssh2_auth_password()` : Identification via SSH en utilisant un mot de passe en clair

bool `ssh2_auth_password` (resource session , string username , string password)

`ssh2_auth_password` tente une identification via SSH en utilisant un mot de passe en clair.

Identification avec un mot de passe

```
<?php
$connection = ssh2_connect('shell.example.com', 22);

if (ssh2_auth_password($connection, 'username', 'secret')) {
    echo "Identification réussi !\n";
} else {
    die('Echec de l'identification...');
}
?>
```

8.125.7 ssh2_auth_pubkey_file() : Identification en utilisant une clé publique

bool **ssh2_auth_pubkey_file** (resource session , string username , string pubkeyfile , string privkeyfile , string passphrase)

`ssh2_auth_pubkey_file` tente une identification en utilisant une clé publique lue depuis un fichier. Si `privkeyfile` est crypté (et doit l'être), la phrase-pass doit être fournie.

Identification en utilisant une clé publique
<pre><?php \$connection = ssh2_connect('shell.example.com', 22, array('hostkey'=>'ssh-rsa')); if (ssh2_auth_pubkey_file(\$connection, 'username', '/home/username/.ssh/id_rsa.pub', '/home/username/.ssh/id_rsa', 'secret')) { echo "Identification réussie en utilisant une clé publique\n"; } else { die('Echec de l'identification en utilisant une clé publique'); } ?></pre>

8.125.8 ssh2_connect() : Connexion à un serveur SSH

resource **ssh2_connect** (string host , int port , array methods , array callbacks)

`ssh2_connect` établit une connexion à un serveur SSH distant et retourne une ressource de connexion en cas de succès, FALSE sinon.

`methods` doit être un tableau associatif avec plus de quatre paramètres, comme décrit ci-dessous.

Index	Signification	Valeurs supportées *
kex	La liste des méthodes d'échange à annoncer, séparées par une virgule, par ordre de préférence.	diffie-hellman-group1-sha1 , diffie-hellman-group14-sha1 , et diffie-hellman-group-exchange-sha1
hostkey	La liste des méthodes de clés d'hôte à annoncer, séparées par une virgule, par ordre de préférence.	ssh-rsa et ssh-dss
client_to_server	Tableau associatif contenant les codes des méthodes de cryptage, de compression et de messages d'identification (MAC) préférés pour l'envoi des messages depuis le client vers le serveur.	
server_to_client	Tableau associatif contenant les codes des méthodes de cryptage, de compression et de messages d'identification (MAC) préférés pour l'envoi des messages depuis le serveur vers le client.	

* - Les valeurs supportées sont dépendantes des méthodes supportées par la bibliothèque. Voir la documentation [libssh2](#) pour plus d'informations.

Index	Signification	Valeurs supportées *
crypt	Liste des méthodes de cryptage à annoncer, séparées par une virgule, par ordre de préférence.	rijndael-cbc@lysator.liu.se , aes256-cbc , aes192-cbc , aes128-cbc , 3des-cbc , blowfish-cbc , cast128-cbc , arcfour , et none**
comp	Liste des méthodes de compression à annoncer, séparées par une virgule, par ordre de préférence.	zlib et none
mac	Liste des méthodes MAC à annoncer, séparées par une virgule, par ordre de préférence.	hmac-sha1 , hmac-sha1-96 , hmac-ripemd160 , hmac-ripemd160@openssh.com , et none**

Note

Crypt and MAC method " none "

Pour des raisons de sécurité, none est désactivé par la bibliothèque `libssh2` à moins qu'il soit activé explicitement durant la compilation en utilisant les options appropriées du `./configure`. Voir la documentation sur la bibliothèque pour plus d'informations.

Index	Signification	Prototype
ignore	Nom de la fonction à appeler lorsqu'un paquet SSH2_MSG_IGNORE est reçu	void ignore_cb(\$message)
debug	Nom de la fonction à appeler lorsqu'un paquet SSH2_MSG_DEBUG est reçu	void debug_cb(\$message, \$language, \$always_display)
macerror	Nom de la fonction à appeler lorsqu'un paquet est reçu mais que le code message d'identification échoue. Si le callback retourne TRUE , l'erreur sera ignorée, sinon, la connexion se terminera.	bool macerror_cb(\$packet)
disconnect	Nom de la fonction à appeler lorsqu'un paquet SSH2_MSG_DISCONNECT est reçu	void disconnect_cb(\$reason, \$message, \$language)

Ouverture d'une connexion forçant 3des-cbc lors de l'envoi de paquets, n'importe quel force aes cipher lors de la réception de paquets, aucune compression dans les deux directions, et change de clés Group1.

```
<?php
/* Notification à l'utilisateur si le serveur termine la connexion */
function my_ssh_disconnect($reason, $message, $language) {
    printf("Le serveur s'est déconnecté avec le code [%d] et le message : %s\n",
        $reason, $message);
}

$methods = array(
    'kex' => 'diffie-hellman-group1-sha1',
    'client_to_server' => array(
        'crypt' => '3des-cbc',
        'comp' => 'none'),
    'server_to_client' => array(
        'crypt' => 'aes256-cbc,aes192-cbc,aes128-cbc',
        'comp' => 'none'));

$callbacks = array('disconnect' => 'my_ssh_disconnect');

$connection = ssh2_connect('shell.example.com', 22, $methods, $callbacks);
if (!$connection) die('Echec de la connexion');
?>
```

Une fois connecté, le client doit vérifier la clé hôte du serveur en utilisant la fonction `ssh2_fingerprint`, puis s'identifier en utilisant soit un mot de passe, soit une clé publique.

Voir aussi [ssh2_fingerprint](#) , [ssh2_auth_none](#) , [ssh2_auth_password](#) et [ssh2_auth_pubkey_file](#) .

8.125.9 ssh2_exec() : Exécute une commande sur un serveur distant

resource **ssh2_exec** (resource session , string command , string pty , array env , int width , int height , int width_height_type)

[ssh2_exec](#) exécute une commande sur le serveur distant et lui alloue un canal. Retourne un flux en cas de succès, FALSE sinon.

Exécution d'une commande
<pre><?php \$connection = ssh2_connect('shell.example.com', 22); ssh2_auth_password(\$connection, 'username', 'password'); \$stream = ssh2_exec(\$connection, '/usr/local/bin/php -i'); ?></pre>

Voir aussi [ssh2_connect](#) , [ssh2_shell](#) et [ssh2_tunnel](#) .

8.125.10 ssh2_fetch_stream() : Parcours un flux étendu de données

resource **ssh2_fetch_stream** (resource channel , int streamid)

[ssh2_fetch_stream](#) parcourt un sous-flux alternatif associé à un flux de canal SSH2.

[ssh2_fetch_stream](#) parcourt un sous-flux alternatif associé à un flux de canal SSH2 identifié par streamid . Le protocole SSH2 définit actuellement un seul sous-flux, STDERR, qui a un identifiant de sous-flux de SSH2_STREAM_STDERR (défini à 1).

Ouverture d'un shell et récupération du flux stderr qui lui est associé.
<pre><?php \$connection = ssh2_connect('shell.example.com', 22); ssh2_auth_password(\$connection, 'username', 'password'); \$stdio_stream = ssh2_shell(\$connection); \$stderr_stream = ssh2_fetch_stream(\$stdio_stream, SSH2_STREAM_STDERR); ?></pre>

Voir aussi [ssh2_shell](#) , [ssh2_exec](#) et [ssh2_connect](#) .

8.125.11 ssh2_fingerprint() : Récupère l'empreinte d'un serveur distant

string **ssh2_fingerprint** (resource session , int flags)

[ssh2_fingerprint](#) retourne l'empreinte d'un serveur distant depuis une session active ; par défaut en une empreinte MD5 encodée en valeurs hexadécimales ASCII.

flags peut être soit SSH2_FINGERPRINT_MD5 ou SSH2_FINGERPRINT_SHA1 associé logiquement avec SSH2_FINGERPRINT_HEX ou SSH2_FINGERPRINT_RAW . Par défaut, vaut SSH2_FINGERPRINT_MD5 | SSH2_FINGERPRINT_HEX .

Comparaison d'une empreinte à une valeur connue

```
<?php
$known_host = '6F89C2F0A719B30CC38ABDF90755F2E4';

$connection = ssh2_connect('shell.example.com', 22);

$fingerprint = ssh2_fingerprint($connection,
    SSH2_FINGERPRINT_MD5 | SSH2_FINGERPRINT_HEX);

if ($fingerprint != $known_host) {
    die("La clé hôte ne correspond pas !\n" .
        "Attaque Man-In-The-Middle possible ?");
}
?>
```

8.125.12 `ssh2_methods_negotiated()` : Retourne une liste de méthodes négociées

array `ssh2_methods_negotiated` (resource session)

`ssh2_methods_negotiated` retourne une liste de méthodes négociées.

Détermine quelles méthodes ont été négociées

```
<?php
$connection = ssh2_connect('shell.example.com', 22);
$methods = ssh2_methods_negotiated($connection);

echo "Clé de cryptage négociée en utilisant : {$methods['kex']}\n";
echo "Identification du serveur en utilisant {$methods['hostkey']}\n";
echo "Empreinte : " . ssh2_fingerprint($connection) . "\n";

echo "Méthodes de transmission des paquets client vers serveur :\n";
echo "\tCrypt: {$methods['client_to_server']['crypt']}\n";
echo "\tComp: {$methods['client_to_server']['comp']}\n";
echo "\tMAC: {$methods['client_to_server']['mac']}\n";

echo "Méthodes de transmission des paquets serveur vers client :\n";
echo "\tCrypt: {$methods['server_to_client']['crypt']}\n";
echo "\tComp: {$methods['server_to_client']['comp']}\n";
echo "\tMAC: {$methods['server_to_client']['mac']}\n";

?>
```

Voir aussi [ssh2_connect](#) .

8.125.13 `ssh2_publickey_add()` : Ajoute une clé publique autorisée

bool `ssh2_publickey_add` (resource pkey , string algoname , string blob , bool *overwrite* , array *attributes*)

Note

Le `publickey` subsystem est utilisé pour gérer les clés publiques sur un serveur sur lequel le client est **déjà** identifié. Pour s'identifier à un system distant en utilisant l'identification par clé publique, utilisez la fonction `ssh2_auth_pubkey_file` à la place.

8.125.14 `ssh2_publickey_init()` : Initialise un Publickey Subsystem (sous-système de clé publique)

resource `ssh2_publickey_init` (resource session)

Demande le subsystem de la clé publique à partir d'un serveur SSH2 déjà connecté.

Le publickey subsystem permet à un client déjà connecté et identifié de gérer la liste des clés publiques autorisées enregistrée sur le serveur cible dans une manière d'implémentation agnostique. Si le serveur ne supporte pas le publickey subsystem, la fonction `ssh2_publickey_init` retournera FALSE .

Note

Le publickey subsystem est utilisé pour gérer les clés publiques sur un serveur sur lequel le client est **déjà** identifié. Pour s'identifier à un system distant en utilisant l'identification par clé publique, utilisez la fonction `ssh2_auth_pubkey_file` à la place.

8.125.15 `ssh2_publickey_list()` : Liste les clés publiques actuellement autorisées

array `ssh2_publickey_list` (resource pkey)

Note

Le publickey subsystem est utilisé pour gérer les clés publiques sur un serveur sur lequel le client est **déjà** identifié. Pour s'identifier à un system distant en utilisant l'identification par clé publique, utilisez la fonction `ssh2_auth_pubkey_file` à la place.

8.125.16 `ssh2_publickey_remove()` : Supprime un clé publique autorisée

bool `ssh2_publickey_remove` (resource pkey , string algoname , string blob)

Note

Le publickey subsystem est utilisé pour gérer les clés publiques sur un serveur sur lequel le client est **déjà** identifié. Pour s'identifier à un system distant en utilisant l'identification par clé publique, utilisez la fonction `ssh2_auth_pubkey_file` à la place.

8.125.17 `ssh2_scp_recv()` : Demande un fichier via SCP

bool `ssh2_scp_recv` (resource session , string remote_file , string local_file)

`ssh2_scp_recv` copie un fichier depuis un serveur distant vers le système de fichiers local.

Téléchargement d'un fichier via SCP

```
<?php
$connection = ssh2_connect('shell.example.com', 22);
ssh2_auth_password($connection, 'username', 'password');

ssh2_scp_recv($connection, '/remote/filename', '/local/filename');
?>
```

Voir aussi `ssh2_scp_send` et `copy` .

8.125.18 `ssh2_scp_send()` : Envoie un fichier via SCP

bool `ssh2_scp_send` (resource `session` , string `local_file` , string `remote_file` , int `create_mode`)

`ssh2_scp_send` copie un fichier depuis le système de fichiers local vers un serveur distant.

`ssh2_scp_send` copie un fichier depuis le système de fichiers local vers un serveur distant. Le fichier sera créé avec le mode spécifié par `create_mode` .

Chargement d'un fichier via SCP

<pre><?php \$connection = ssh2_connect('shell.example.com', 22); ssh2_auth_password(\$connection, 'username', 'password'); ssh2_scp_send(\$connection, '/local/filename', '/remote/filename', 0644); ?></pre>
--

Voir aussi `ssh2_scp_recv` et `copy` .

8.125.19 `ssh2_sftp_lstat()` : Statue un lien symbolique

array `ssh2_sftp_lstat` (resource `sftp` , string `path`)

`ssh2_sftp_lstat` statue un lien symbolique sur le système de fichiers distant **sans** suivre le lien.

`ssh2_sftp_lstat` statue un lien symbolique sur le système de fichiers distant **sans** suivre le lien. Cette fonction est similaire à l'utilisation de la fonction `lstat` avec le gestionnaire `ssh2.sftp://` en PHP 5 et retourne les mêmes valeurs. Voir la documentation de la fonction `stat` pour les détails concernant les valeurs retournées.

Statue un lien symbolique via SFTP

<pre><?php \$connection = ssh2_connect('shell.example.com', 22); ssh2_auth_password(\$connection, 'username', 'password'); \$sftp = ssh2_sftp(\$connection); \$stainfo = ssh2_lstat(\$sftp, '/path/to/symlink'); \$filesize = \$stainfo['size']; \$group = \$stainfo['gid']; \$owner = \$stainfo['uid']; \$atime = \$stainfo['atime']; \$mtime = \$stainfo['mtime']; \$mode = \$stainfo['mode']; ?></pre>

Voir aussi `ssh2_sftp_stat` , `lstat` et `stat` .

8.125.20 `ssh2_sftp_mkdir()` : Crée un dossier

bool `ssh2_sftp_mkdir` (resource `sftp` , string `dirname` , int `mode` , bool `recursive`)

`ssh2_sftp_mkdir` crée un dossier sur le système de fichiers distant.

`ssh2_sftp_mkdir` crée un dossier sur le système de fichiers distant avec les permissions définies à mode . Si recursive vaut TRUE , tous les dossiers requis pour dirname seront également automatiquement créés. Cette fonction est similaire à la fonction `mkdir` avec le gestionnaire `ssh2.sftp://` .

Création d'un dossier sur un serveur distant

```
<?php
$connection = ssh2_connect('shell.example.com', 22);
ssh2_auth_password($connection, 'username', 'password');
$sftp = ssh2_sftp($connection);

ssh2_sftp_mkdir($sftp, '/home/username/newdir');
/* Ou : mkdir("ssh2.sftp://$sftp/home/username/newdir"); */
?>
```

Voir aussi `mkdir` et `ssh2_sftp_rmdir` .

8.125.21 `ssh2_sftp_readlink()` : Retourne la cible d'un lien symbolique

string `ssh2_sftp_readlink` (resource sftp , string link)

`ssh2_sftp_readlink` retourne la cible d'un lien symbolique.

Lecture d'un lien symbolique

```
<?php
$connection = ssh2_connect('shell.example.com', 22);
ssh2_auth_password($connection, 'username', 'password');
$sftp = ssh2_sftp($connection);

$target = ssh2_sftp_readlink($sftp, '/tmp/mysql.sock');
/* $target est maintenant (e.g.): '/var/run/mysql.sock' */
?>
```

Voir aussi `readlink` et `ssh2_sftp_symlink` .

8.125.22 `ssh2_sftp_realpath()` : Résoud le chemin réel d'un chemin fourni

string `ssh2_sftp_realpath` (resource sftp , string filename)

`ssh2_sftp_realpath` traduit un nom de fichier filename en son chemin réellement effectif sur le système de fichiers distant.

Résoudre un nom de chemin

```
<?php
$connection = ssh2_connect('shell.example.com', 22);
ssh2_auth_password($connection, 'username', 'password');
$sftp = ssh2_sftp($connection);

$realpath = ssh2_sftp_realpath($sftp, '/home/username/../../../../usr/../etc/passwd');
/* $realpath est maintenant : '/etc/passwd' */
?>
```

Voir aussi `realpath` , `ssh2_sftp_symlink` et `ssh2_sftp_readlink` .

8.125.23 `ssh2_sftp_rename()` : Renomme un fichier distant

bool `ssh2_sftp_rename` (resource `sftp` , string `from` , string `to`)

`ssh2_sftp_rename` renomme un fichier sur le système de fichier distant.

Renommer un fichier via sftp

```

<?php
$connection = ssh2_connect('shell.example.com', 22);
ssh2_auth_password($connection, 'username', 'password');
$sftp = ssh2_sftp($connection);

ssh2_sftp_rename($sftp, '/home/username/oldname', '/home/username/newname');
?>

```

Voir aussi [rename](#) .

8.125.24 `ssh2_sftp_rmdir()` : Efface un dossier

bool `ssh2_sftp_rmdir` (resource `sftp` , string `dirname`)

`ssh2_sftp_rmdir` efface un dossier depuis le système de fichiers distant.

`ssh2_sftp_rmdir` efface un dossier depuis le système de fichiers distant. Cette fonction est similaire à l'utilisation de la fonction `rmdir` avec le gestionnaire `ssh2.sftp://` .

Effacement d'un dossier sur un serveur distant

```

<?php
$connection = ssh2_connet('shell.example.com', 22);
ssh2_auth_password($connection, 'username', 'password');
$sftp = ssh2_sftp($connection);

ssh2_sftp_rmdir($sftp, '/home/username/deltodel');
/* Ou : rmdir("ssh2.sftp://$sftp/home/username/dirtodel"); */
?>

```

Voir aussi [rmdir](#) et [ssh2_sftp_mkdir](#) .

8.125.25 `ssh2_sftp_stat()` : Statue un fichier sur un système de fichiers distant

array `ssh2_sftp_stat` (resource `sftp` , string `path`)

`ssh2_sftp_stat` statue un fichier distant. Si le fichier est un lien symbolique, la fonction suivra le lien et statuera la cible.

`ssh2_sftp_stat` statue un fichier distant. Si le fichier est un lien symbolique, la fonction suivra le lien et statuera la cible. Cette fonction est similaire à l'utilisation de la fonction `stat` avec le gestionnaire `ssh2.sftp://` en PHP 5 et retourne les mêmes valeurs. Voir la documentation de la fonction `stat` pour les détails concernant les valeurs retournées.

Statut d'un fichier via SFTP

```

<?php
$connection = ssh2_connect('shell.example.com', 22);

```

```

ssh2_auth_password($connection, 'username', 'password');

$sftp = ssh2_sftp($connection);
$stathinfo = ssh2_stat($sftp, '/path/to/symlink');

$filesize = $stathinfo['size'];
$group = $stathinfo['gid'];
$owner = $stathinfo['uid'];
$atime = $stathinfo['atime'];
$mtime = $stathinfo['mtime'];
$mode = $stathinfo['mode'];
?>

```

Voir aussi [ssh2_sftp_lstat](#) , [lstat](#) et [stat](#) .

8.125.26 ssh2_sftp_symlink() : Crée un lien symbolique

bool **ssh2_sftp_symlink** (resource sftp , string target , string link)

[ssh2_sftp_symlink](#) crée un lien symbolique sur le système de fichiers distant.

[ssh2_sftp_symlink](#) crée un lien symbolique sur le système de fichiers distant pointant vers target .

Création d'un lien symbolique

```

<?php
$connection = ssh2_connect('shell.example.com', 22);
ssh2_auth_password($connection, 'username', 'password');
$sftp = ssh2_sftp($connection);

ssh2_sftp_symlink($sftp, '/var/run/mysql.sock', '/tmp/mysql.sock');
?>

```

Voir aussi [symlink](#) et [ssh2_sftp_readlink](#) .

8.125.27 ssh2_sftp_unlink() : Efface un fichier

bool **ssh2_sftp_unlink** (resource sftp , string filename)

[ssh2_sftp_unlink](#) efface un fichier sur le système de fichier distant.

Effacement d'un fichier

```

<?php
$connection = ssh2_connect('shell.example.com', 22);
ssh2_auth_password($connection, 'username', 'password');
$sftp = ssh2_sftp($connection);

ssh2_sftp_unlink($sftp, '/home/username/stale_file');
?>

```

Voir aussi [unlink](#) .

8.125.28 ssh2_sftp() : Initialise un sous-système SFTP

resource **ssh2_sftp** (resource session)

[ssh2_sftp](#) demande un sous-système SFTP depuis un serveur déjà connecté SSH2.

Cette méthode retourne une ressource SSH2 SFTP pour utilisation avec toutes les méthodes [ssh2_sftp_*\(\)](#) ainsi que le gestionnaire ouvert [ssh2.sftp://](#) .

Ouverture d'un fichier via SFTP

```
<?php
$connection = ssh2_connect('shell.example.com', 22);
ssh2_auth_password($connection, 'username', 'password');

$sftp = ssh2_sftp($connection);

$stream = fopen("ssh2.sftp://$sftp/path/to/file", 'r');
?>
```

Voir aussi [ssh2_scp_send](#) et [ssh2_scp_recv](#) .

8.125.29 ssh2_shell() : Demande un shell interactif

resource **ssh2_shell** (resource session , string term_type , array env , int width , int height , int width_height_type)

[ssh2_shell](#) ouvre un shell sur le serveur distant et lui alloue un flux.

[ssh2_shell](#) ouvre un shell sur le serveur distant et lui alloue un flux. `term_type` doit correspondre à une des entrées du fichier `/etc/termcap` du système cible et vaut par défaut `vanilla` . `env` doit être passé en tant qu'un tableau associatif de paire nom/valeur à définir dans l'environnement cible.

`width` et `height` définissent la hauteur et la largeur du terminal virtuel alloué au processus du shell. `width_height_type` doit être soit `SSH2_TERM_UNIT_CHARS` , soit `SSH2_TERM_UNIT_PIXELS` .

Exécution d'une commande

```
<?php
$connection = ssh2_connect('shell.example.com', 22);
ssh2_auth_password($connection, 'username', 'password');

$stream = ssh2_shell($connection, 'vt102', null, 80, 24, SSH2_TERM_UNIT_CHARS);
?>
```

Voir aussi [ssh2_exec](#) , [ssh2_tunnel](#) et [ssh2_fetch_stream](#) .

8.125.30 ssh2_tunnel() : Ouvre un tunnel à travers un serveur distant

resource **ssh2_tunnel** (resource session , string host , int port)

[ssh2_tunnel](#) ouvre un socket vers un hôte/port arbitraire via un serveur connecté SSH.

Ouverture d'un tunnel sur un hôte arbitraire

```
<?php
$connection = ssh2_connect('shell.example.com', 22);
ssh2_auth_pubkey_file($connection, 'username', 'id_dsa.pub', 'id_dsa');

$tunnel = ssh2_tunnel($connection, '10.0.0.101', 12345);
?>
```

Voir aussi [ssh2_connect](#) et [fsockopen](#) .

8.126 Flux

8.126.1 Introduction

Les flux (" streams " en anglais) ont été introduits en PHP 4.3.0 comme méthode de généralisation des fichiers, sockets, connexions réseau, données compressées et autres opérations du même type, qui partagent des opérations communes. Dans sa définition la plus simple, un flux est une ressource qui présente des capacités de flux : c'est-à-dire que ces objets peuvent être lus ou recevoir des écritures de manière linéaire, et dispose aussi de moyen d'accéder à des positions arbitraires dans le flux.

Un gestionnaire (dit wrapper en anglais), est une fonction qui indique comment le flux se comporte spécifiquement. C'est le cas du gestionnaire http , qui sait comment traduire une URL en une requête HTTP/1.0 sur un serveur distant. Il existe de nombreux gestionnaires intégrés à PHP par défaut (voir [Liste des protocoles supportés](#)), et, de plus, des gestionnaires spécifiques peuvent être ajoutés dans les scripts PHP avec la fonction `stream_register_wrapper` , ou bien directement par une autre extension, en utilisant l'API C de [API de flux : les auteurs d'extensions](#) . Grâce à la souplesse des gestionnaires qui peuvent être ajoutés à PHP, il n'y a pas de limites aux possibilités offertes. Pour connaître la liste des gestionnaires actuellement enregistrés, utilisez la fonction `stream_get_wrappers` .

Un flux est référencé comme : `scheme :// target`

- `scheme` ([chaîne de caractères](#)) - Le nom du gestionnaire à utiliser. Par exemple, `file` , `http` , `https` , `ftp` , `ftps` , `compress.zlib` , `compress.bz` . et `php` . Voir [Liste des protocoles supportés](#) pour une liste complète des gestionnaires enregistrés de PHP. Si aucun gestionnaire n'est spécifié, la fonction par défaut est utilisée (typiquement, `file ://`).
- `target` - Dépend du gestionnaire utilisé. Pour les flux relatifs aux systèmes de fichiers, c'est typiquement un chemin et un nom de fichier du fichier désiré. Pour les flux relatifs aux réseaux, c'est typiquement le nom d'hôte, souvent avec un chemin apposé. Voir aussi [Liste des protocoles supportés](#) pour une description des cibles des flux intégrés.

8.126.2 Filtres de flux

Un filtre est une fonction finale qui effectue des opérations sur les données qui sont lues ou écrites dans un flux. Un nombre arbitraire de filtres peuvent être ajoutés sur un flux. Des filtres personnalisés peuvent aussi être ajoutés avec la fonction `stream_register_filter` , ou bien dans une extension avec l'API C de [API de flux : les auteurs d'extensions](#) . Pour connaître la liste des gestionnaires actuellement enregistrés, utilisez la fonction `stream_get_filters` .

8.126.3 Contextes de flux

Un contexte est un jeu de paramètres et d'options spécifiques à un gestionnaire qui modifie ou améliore le comportement d'un flux. Les contextes sont créés en utilisant la fonction `stream_context_create` et peuvent être donnés aux fonctions de création de flux sur le système de fichiers (i.e. `fopen` , `file` , `file_get_contents` , etc.).

Les options peuvent être spécifiées en appelant `stream_context_create` ou, plus tard, avec `stream_context_set_option` . Une liste des options spécifiques à des gestionnaires est disponible dans la liste des gestionnaires intégrés (voyez [Liste des protocoles supportés](#)).

De plus, les paramètres peuvent être envoyés à un contexte en utilisant la fonction [stream_context_set_params](#) . Actuellement, le seul paramètre de contexte supporté par PHP est notification . La valeur de ce paramètre doit être le nom d'une fonction qui sera appelée lorsqu'un événement survient pour un flux. La fonction d'alerte est appelée durant la réception de l'événement, et doit accepter 6 paramètres :

```
void my_notifier ( int notification_code , int severity , string message , int message_code , int bytes_transferred , int bytes_max )
```

notification_code et severity sont des valeurs numériques qui correspondent aux constantes STREAM_NOTIFY_* listées ci-dessous. Si un message descriptif est disponible dans un flux, les paramètres message et message_code en seront équipés. La signification de ces valeurs est dépendante du gestionnaire. bytes_transferred et bytes_max seront fournies si possible.

8.126.4 Installation

Les flux font partie de PHP depuis la version 4.3.0. Aucune étape supplémentaire n'est requise pour les activer.

8.126.5 Classes Stream

Des gestionnaires personnalisés peuvent être enregistrés via la fonction [stream_register_wrapper](#) , en utilisant la définition de classe décrite dans ce manuel.

La classe php_user_filter est prédéfinie. C'est une classe abstraite à utiliser avec les filtres personnalisés. Voyez le manuel de la fonction [stream_register_filter](#) pour plus de détails sur les implémentations de filtres utilisateurs.

8.126.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Constante	Description
STREAM_FILTER_READ	Utilisée avec stream_filter_append et stream_filter_prepend pour indiquer que le filtre spécifié ne doit être appliqué qu'après la lecture .
STREAM_FILTER_WRITE	Utilisée avec stream_filter_append et stream_filter_prepend pour indiquer que le filtre spécifié ne doit être appliqué qu'après l' écriture .
STREAM_FILTER_ALL	Cette constante est équivalente à STREAM_FILTER_READ STREAM_FILTER_WRITE
PSFS_PASS_ON *	Le code retourné indique que le filtre utilisateur retourne des données dans \$out .
PSFS_FEED_ME *	Le code retourné indique que le filtre utilisateur ne retourne pas de données dans \$out (i.e. Aucune donnée disponible).
PSFS_ERR_FATAL *	Le code retourné indique que le filtre utilisateur a produit une erreur fatale. (i.e. Données invalides reçues).

STREAM_USE_PATH	Option indiquant si stream a utilisé l' <code>include_path</code> .
STREAM_REPORT_ERRORS	Option indiquant si le gestionnaire est responsable pour la levée des erreurs avec <code>trigger_error</code> durant l'ouverture du flux. Si cette constante n'existe pas, vous ne devez pas émettre d'erreurs.
STREAM_CLIENT_ASYNC_CONNECT *	Ouvre un socket client en mode asynchrone. Cette option doit être utilisée avec le flag <code>STREAM_CLIENT_CONNECT</code> . A utiliser avec la fonction <code>stream_socket_client</code> .
STREAM_CLIENT_CONNECT *	Ouvre un socket client. Les sockets clients doivent toujours inclure ce flag. A utiliser avec la fonction <code>stream_socket_client</code> .
STREAM_CLIENT_PERSISTENT *	Le socket client ouvert avec <code>stream_socket_client</code> doit rester persistant entre chaque page chargée.
STREAM_SERVER_BIND *	Appel un flux créé avec la fonction <code>stream_socket_server</code> pour s'identifier sur la cible définie. Les sockets serveur doivent toujours utiliser cette constante.
STREAM_SERVER_LISTEN *	Appel un flux créé avec <code>stream_socket_server</code> et utilise la constante <code>STREAM_SERVER_BIND</code> pour commencer à écouter la socket. Les connexions orientées transports (comme TCP) doivent utiliser ce flag sinon la socket serveur ne sera pas activé. Utiliser ce flag pour les connexions basses de transports (comme UDP) est une erreur.
STREAM_NOTIFY_RESOLVE *	Une adresse distante requise pour ce flux a été résolue, ou bien la résolution a échoué. Voir le paramètre <code>severity</code> pour avoir une indication sur l'événement survenu.
STREAM_NOTIFY_CONNECT	Une connexion avec une ressource externe a été établie.
STREAM_NOTIFY_AUTH_REQUIRED	Une autorisation supplémentaire est demandée pour accéder à la ressource spécifiée. Typiquement utilisé avec le niveau d'alerte <code>severity</code> de la constante <code>STREAM_NOTIFY_SEVERITY_ERR</code> .
STREAM_NOTIFY_MIME_TYPE_IS	Le type mime de la ressource a été identifié : voir le paramètre <code>message</code> pour une description du type découvert.
STREAM_NOTIFY_FILE_SIZE_IS	La taille de la ressource a été découverte.
STREAM_NOTIFY_REDIRECTED	La ressource externe a redirigé le flux vers un endroit différent. Voir le paramètre <code>message</code> .
STREAM_NOTIFY_PROGRESS	Indique l'actuelle progression du transfert du flux dans <code>bytes_transferred</code> et peut-être <code>bytes_max</code> également.
STREAM_NOTIFY_COMPLETED *	Il n'y a plus de données disponibles sur le flux.
STREAM_NOTIFY_FAILURE	Une erreur générique est intervenue sur le flux, consultez les paramètres <code>message</code> et <code>message_code</code> pour plus de détails.
STREAM_NOTIFY_AUTH_RESULT	L'autorisation est terminée (avec succès ou pas).
STREAM_NOTIFY_SEVERITY_INFO	Notification normale, aucune erreur signalée.
STREAM_NOTIFY_SEVERITY_WARN	Erreur non critique. Le traitement continue.
STREAM_NOTIFY_SEVERITY_ERR	Une erreur critique est survenu. Le traitement ne peut continuer.
STREAM_IPPROTO_ICMP +	Fournit un socket ICMP.

STREAM_IPPROTO_IP +	Fournit un socket IP.
STREAM_IPPROTO_RAW +	Fournit un socket RAW.
STREAM_IPPROTO_TCP +	Fournit un socket TCP.
STREAM_IPPROTO_UDP +	Fournit un socket UDP.
STREAM_PF_INET +	Protocole Internet version 4 (IPv4).
STREAM_PF_INET6 +	Protocole internet version 6 (IPv6).
STREAM_PF_UNIX +	Protocoles internes des systèmes Unix.
STREAM SOCK_DGRAM +	Fournit des datagrammes, qui sont des messages de connexion (UDP, par exemple).
STREAM SOCK_RAW +	Fournit un socket raw, qui fournit un accès aux protocoles et interfaces internes du réseau. Habituellement, ce type de socket n'est disponible qu'à l'utilisateur root.
STREAM SOCK_RDM +	Fournit un socket RDM (Reliably-delivered messages).
STREAM SOCK_SEQPACKET +	Fournit un socket de flux de paquets séquencés.
STREAM SOCK_STREAM +	Fournit un flux séquencé, deux chemins avec un mécanisme de transmission pour les données "out-of-band" (TCP par exemple).
Note	
Les constantes marquées avec une * sont uniquement disponibles depuis PHP 5.0.0.	
Note	
Les constantes marquées avec une + sont disponibles depuis PHP 5.1.0 et sont faites pour être utilisées avec la fonction <code>stream_socket_pair</code> . Notez que quelques-unes de ces constantes peuvent ne pas être disponibles sur votre système.	

8.126.7 Erreurs de flux

Comme avec n'importe quel fichier ou socket, les opérations sur un flux peuvent échouer pour une grande variété de raisons (par exemple : impossible de se connecter au serveur distant, fichier introuvable, etc.). Un flux peut aussi échouer parce que le gestionnaire n'est pas configuré sur le système en cours. Voyez le tableau retourné par la fonction `stream_get_wrappers` pour connaître la liste des gestionnaires configurés sur votre installation de PHP. Comme avec la plupart des fonctions internes de PHP, si une erreur survient, un message de type `E_WARNING` sera généré pour indiquer la nature de l'erreur.

8.126.8 Exemples

Exemples avec <code>file_get_contents</code>
<pre><?php /* Lit un fichier local dans le dossier /home/bar */ \$localfile = file_get_contents("/home/bar/foo.txt"); /* Identique au précédent, mais utilise explicitement le gestionnaire FILE */ \$localfile = file_get_contents("file:///home/bar/foo.txt"); /* Lit un fichier distant sur le serveur www.example.com avec le protocole HTTP */ \$httpfile = file_get_contents("http://www.example.com/foo.txt"); /* Lit le même fichier sur le serveur www.example.com avec le protocole HTTPS */ \$httpsfile = file_get_contents("https://www.example.com/foo.txt"); /* Lit un fichier distant sur le serveur ftp.example.com en utilisant le protocole FTP */ \$ftpfile = file_get_contents("ftp://user:pass@ftp.example.com/foo.txt");</pre>

```
/* Lit un fichier distant sur le serveur ftp.example.com en utilisant le protocole FTPS */
$ftpsfile = file_get_contents("ftps://user:pass@ftp.example.com/foo.txt");
?>
```

Envoi d'une requête de type POST sur un serveur sécurisé

```
<?php
/* Envoi d'une requête POST sur le serveur https://secure.example.com/form_action.php
 * Inclusion des variables "foo" et "bar"
 */

$sock = fsockopen("ssl://secure.example.com", 443, $errno, $errstr, 30);
if (!$sock) die("$errstr ($errno)\n");

$data = "foo=" . urlencode("Valeur de Foo") . "&bar=" . urlencode("Valeur de Bar");

fputs($sock, "POST /form_action.php HTTP/1.0\r\n");
fputs($sock, "Host: secure.example.com\r\n");
fputs($sock, "Content-type: application/x-www-form-urlencoded\r\n");
fputs($sock, "Content-length: " . strlen($data) . "\r\n");
fputs($sock, "Accept: */*\r\n");
fputs($sock, "\r\n");
fputs($sock, "$data\r\n");
fputs($sock, "\r\n");

$headers = "";
while ($str = trim(fgets($sock, 4096))) {
    $headers .= "$str\n";
}

echo "\n";

$body = "";
while (!feof($sock)) {
    $body .= fgets($sock, 4096);
}

fclose($sock);
?>
```

Ecrire des données dans un fichier compressé

```
<?php
/* Création d'un fichier compressé contenant une chaîne arbitraire
 * Le fichier peut être lu en utilisant le gestionnaire compress.zlib
 * ou simplement décompressé; en ligne de commande avec 'gzip -d foo-bar.txt.gz'
 */
$fp = fopen("compress.zlib://foo-bar.txt.gz", "w");
if (!$fp) die("Impossible de créer le fichier.");

fwrite($fp, "Ceci est un test.\n");

fclose($fp);
?>
```

- [Introduction](#)
- [Filtres de flux](#)
- [Contextes de flux](#)
- [Installation](#)
- [Classes Stream](#)
- [Constantes pré-définies](#)

- [Erreurs de flux](#)
- [Exemples](#)
- [stream_bucket_append](#)
- [stream_bucket_make_writeable](#)
- [stream_bucket_new](#)
- [stream_bucket_prepend](#)
- [stream_context_create](#)
- [stream_context_get_default](#)
- [stream_context_get_options](#)
- [stream_context_set_option](#)
- [stream_context_set_params](#)
- [stream_copy_to_stream](#)
- [stream_filter_append](#)
- [stream_filter_prepend](#)
- [stream_filter_register](#)
- [stream_filter_remove](#)
- [stream_get_contents](#)
- [stream_get_filters](#)
- [stream_get_line](#)
- [stream_get_meta_data](#)
- [stream_get_transports](#)
- [stream_get_wrappers](#)
- [stream_register_wrapper](#)
- [stream_select](#)
- [stream_set_blocking](#)
- [stream_set_timeout](#)
- [stream_set_write_buffer](#)
- [stream_socket_accept](#)
- [stream_socket_client](#)
- [stream_socket_enable_crypto](#)
- [stream_socket_get_name](#)
- [stream_socket_pair](#)
- [stream_socket_recvfrom](#)
- [stream_socket_sendto](#)
- [stream_socket_server](#)
- [stream_wrapper_register](#)
- [stream_wrapper_restore](#)
- [stream_wrapper_unregister](#)

8.126.10 [stream_bucket_make_writeable\(\)](#) : Retourne un objet de compartiment depuis le corps pour des opérations sur celui-ci

objet [stream_bucket_make_writeable](#) (resource brigade)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.126.11 [stream_bucket_new\(\)](#) : Crée un nouveau compartiment pour l'utiliser sur le flux courant

objet [stream_bucket_new](#) (resource stream , string buffer)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.126.12 `stream_bucket_prepend()` : Ajout initial d'un compartiment au corps

void `stream_bucket_prepend` (resource brigade , resource bucket)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.126.13 `stream_context_create()` : Crée un contexte de flux

resource `stream_context_create` (array options)

`stream_context_create` crée et retourne un contexte de flux, avec les paramètres fournis par options .

Le paramètre options doit être un tableau associatif, au format `$arr['wrapper']['option'] = $value` . Par défaut, c'est un tableau vide.

Exemple avec <code>stream_context_create</code>

```
<?php
$opts = array (
    'http'=>array (
        'method' => 'GET',
        'header' => "Accept-language: en\r\n" .
                  "Cookie: foo=bar\r\n"
    )
);
$context = stream_context_create($opts);

// Envoie une requête HTTP à www.example.com
// avec des en-têtes supplémentaires
$fp = fopen('http://www.example.com', 'r', false, $context);
fpassthru($fp);
fclose($fp);
?>
```

Voir aussi `stream_context_set_params` et la liste des gestionnaires avec des options de contexte ([Liste des protocoles supportés](#)).

8.126.14 `stream_context_get_default()` : Lit le contexte par défaut des flux

resource `stream_context_get_default` (array options)

`stream_context_get_default` retourne le contexte par défaut qui est utilisé avec les fonctions de fichiers comme `fopen` , `file_get_contents` , etc, lorsqu'elles sont utilisées sans paramètre de contexte. Les options du contexte par défaut peuvent être spécifiées optionnellement avec la même syntaxe que pour `stream_context_create` .

8.126.11 `stream_bucket_new()` : Crée un nouveau compartiment pour l'utiliser sur le flux courant 1852

options doit être un tableau associatif de tableaux associatifs, au format \$arr['wrapper']['option'] = \$value .

Exemple avec `stream_context_get_default`

```
<?php
$default_opts = array(
    'http'=>array(
        'method'=>"GET",
        'header'=>"Accept-language: en\r\n" .
            "Cookie: foo=bar",
        'proxy'=>"tcp://10.54.1.39:8000"
    )
);

$alternate_opts = array(
    'http'=>array(
        'method'=>"POST",
        'header'=>"Content-type: application/x-www-form-urlencoded\r\n" .
            "Content-length: " . strlen("baz=bomb"),
        'content'=>"baz=bomb"
    )
);

$default = stream_context_get_default($default_opts);
$alternate = stream_context_create($alternate_opts);

/* Envoie une requête GET classique à un serveur proxy 10.54.1.39
 * vers www.example.com, en utilisant les options de contexte spécifiées
 * dans $default_opts
 */
readfile('http://www.example.com');

/* Envoie une requête POST directement à www.example.com
 * Utilise les options de contexte de $alternate_opts
 */
readfile('http://www.example.com', false, $alternate);

?>
```

Voir aussi [stream_context_create](#) et les gestionnaires supportés par les contextes ([Liste des protocoles supportés](#)).

8.126.15 `stream_context_get_options()` : Lit la valeur des options pour un flux/gestionnaire/contexte

array `stream_context_get_options` (resource stream_or_context)

`stream_context_get_options` retourne un tableau contenant la liste des options pour le flux ou le contexte stream|context .

8.126.16 `stream_context_set_option()` : Configure une option pour un flux/gestionnaire/contexte

bool `stream_context_set_option` (resource stream_or_context , string wrapper , string option , mixed value) bool `stream_context_set_option` (resource stream_or_context , array options)

stream_context_set_option configure l'option option avec la nouvelle valeur value le flux ou le gestionnaire ou le contexte wrapper .

8.126.17 stream_context_set_params() : Configure les paramètres pour un flux/gestionnaire/contexte

bool **stream_context_set_params** (resource stream_or_context , array params)

stream_context_set_params configure les paramètres pour un flux, un gestionnaire ou un contexte. params doit être un tableau associatif avec la structure suivante : \$params[nom_parametre] = "valeur_parametre"; .

Paramètres	Usage
notification	Nom de la fonction utilisateur appelée dès qu'un flux génère une notification.

8.126.18 stream_copy_to_stream() : Copie des données depuis un flux vers un autre

int **stream_copy_to_stream** (resource source , resource dest , int maxlength , int offset)

Fait une copie jusqu'à maxlength octets de données depuis la position courante du pointeur (ou depuis la position offset , si spécifié) dans le flux source vers le paramètre dest . Si maxlength n'est pas spécifié, tout le reste du flux source sera copié.

8.126.19 stream_filter_append() : Attache un filtre à un flux en fin de liste

resource **stream_filter_append** (resource stream , string filtername , int read_write , mixed params)

stream_filter_append ajoute le filtre filtername à la liste de filtres attachés au flux stream . Ce filtre sera ajouté avec les paramètres spécifiés dans params à la **fin** de cette liste et sera donc appelé en dernier durant les opérations de flux. Pour ajouter un filtre au début de la liste, utilisez la fonction stream_filter_prepend .

Par défaut, stream_filter_append va ajouter le filtre à la liste de filtres de lecture si le fichier a été ouvert en mode lecture (r et/ou +). Le filtre sera aussi attaché à la liste des filtres de lecture si le fichier a été ouvert en mode lecture (w , a et/ou +). STREAM_FILTER_READ , STREAM_FILTER_WRITE , et/ou STREAM_FILTER_ALL peuvent aussi être utilisées dans le paramètre read_write pour contrôler ce comportement. Voyez la fonction stream_filter_append pour un exemple de l'utilisation de ce paramètre.

Depuis PHP 5.0.1, cette fonction retourne une ressource qui peut être utilisée pour se référer à cette instance de filtre durant l'appel à la fonction stream_filter_remove . Dans les versions antérieures à PHP 5.1.0, cette fonction retourne FALSE en cas de succès, FALSE sinon.

Contrôler l'application des filtres
<pre><?php // Ouverture d'un fichier de test en lecture/écriture \$fp = fopen('test.txt', 'w');</pre>

```

/* On applique le filtre ROT13 au flux d'écriture, mais pas à
 * celui de lecture */
stream_filter_append($fp, "string.rot13", STREAM_FILTER_WRITE);

/* On ajoute un simple chaîne dans le fichier, il sera
 * transformé par ROT13 à l'écriture */
fwrite($fp, "Ceci est un test\n");

/* On revient au début du fichier */
rewind($fp);

/* On lit le contenu du fichier.
 * Si on appliquait le filtre ROT13 nous aurions la
 * chaîne dans son état d'origine */
fpassthru($fp);

fclose($fp);

/* Résultat attendu
   -----
Guvf vf n grfg

*/
?>

```

Note

Quand vous utilisez des filtres personnalisés

stream_register_filter doit être appelée avant stream_filter_append pour enregistrer le filtre sous le nom de `filtername` .

Note

Les données du flux (locales et distantes) sont retournées en morceaux, les données non acheminées étant conservées dans le tampon interne. Lorsqu'un nouveau filtre est ajouté à la fin du flux, les données dans le tampon interne sont passées dans le nouveau filtre à ce moment-là. Ceci est différent du comportement de stream_filter_prepend .

Voir aussi stream_filter_register , stream_filter_prepend et stream_get_filters .

8.126.20 stream_filter_prepend() : Attache un filtre à un flux en début de liste

resource **stream_filter_prepend** (resource `stream` , string `filtername` , int `read_write` , mixed `params`)

stream_filter_prepend ajoute le filtre `filtername` à la liste de filtres attachés au flux `stream` . Ce filtre sera ajouté avec les paramètres spécifiés dans `params` à la **fin** de cette liste et sera donc appelé en dernier durant les opérations de flux. Pour ajouter un filtre à la fin de la liste, utilisez la fonction stream_filter_prepend .

Par défaut, stream_filter_prepend va ajouter le filtre à la liste de filtres de lecture si le fichier a été ouvert en mode lecture (`r` et/ou `+`). Le filtre sera aussi attaché à la liste des filtres de lecture si le fichier a été ouvert en mode lecture (`w` , `a` et/ou `+`). `STREAM_FILTER_READ` , `STREAM_FILTER_WRITE` , et/ou `STREAM_FILTER_ALL` peuvent aussi être utilisées dans le paramètre `read_write` pour contrôler ce comportement. Voyez la fonction stream_filter_append pour un exemple de l'utilisation de ce paramètre.

Depuis PHP 5.0.1, cette fonction retourne une ressource qui peut être utilisée pour se référer à cette instance de filtre durant l'appel à la fonction [stream_filter_remove](#) . Dans les versions antérieures à PHP 5.1.0, cette fonction retourne FALSE en cas de succès, FALSE sinon.

Note
Quand vous utilisez des filtres personnalisés stream_register_filter doit être appelée avant stream_filter_prepend pour enregistrer le filtre sous le nom de filtername .
Note
Les données du flux (locales et distantes) sont retournées en morceaux, les données non acheminées étant conservées dans le tampon interne. Lorsqu'un nouveau filtre est ajouté au début du flux, les données dans le tampon interne ne sont pas passées dans le nouveau filtre à ce moment là. Ceci est différent du comportement de stream_filter_append .
Voir aussi stream_register_filter et stream_filter_append .

8.126.21 stream_filter_register() : Enregistre un filtre de flux

bool **stream_filter_register** (string filtername , string classname)

[stream_filter_register](#) vous permet d'implémenter votre propre filtre de flux, à utiliser avec les fonctions d'accès aux données externes (comme [fopen](#) , [fread](#) , etc.).

Pour ce faire, vous devez définir une classe qui étend la classe php_user_filter avec les membres et méthodes définies ci-dessous. Lorsque vous réalisez des opérations de lecture et d'écriture dans le flux auquel votre filtre est attaché, PHP passera les données à travers votre filtre (et tous les autres filtres attachés), de façon à ce que les données soient modifiées telles que désiré. Vous devez implémenter les méthodes telles que décrit ci-dessous, sous peine de comportements indéfinis.

[stream_filter_register](#) retourne FALSE si le nom de filtre filtername est déjà utilisé.

int **filter** (resource in , resource out , int consumed , bool closing)

Cette méthode est appelée à chaque fois que des données sont lues ou écrites dans le flux attaché (avec des fonctions comme [fread](#) ou [fwrite](#)). Le paramètre in est une ressource qui pointe sur une bucket brigade qui contient un ou plusieurs objet bucket contenant les données à filtrer. out est une autre ressource qui pointe sur une bucket brigade dans laquelle les données seront placées. consumed , qui doit **toujours** être déclaré par référence, doit être incrémenté de la taille de données que votre filtre lit et modifie. Dans la plupart des cas, cela signifie que vous devrez incrémenter consumed avec \$bucket->datalen pour chaque \$bucket . Si le flux est en cours de fermeture (et, par conséquent, cela sera le dernier passage dans la chaîne de filtres), le paramètre closing vaudra TRUE La méthode **filter** doit retourner l'une des trois valeurs suivantes :

Valeur retournée	Signification
PSFS_PASS_ON	Indique la réussite ; les données sont dans le paramètre out , une bucket brigade .
PSFS_FEED_ME	Indique que le filtre n'a aucune donnée à retourner, et requiert d'autres données du flux.
PSFS_ERR_FATAL (défaut)	Indique que le flux a rencontré une erreur fatale, et ne peut continuer. Si aucune valeur n'est retournée par cette méthode, PSFS_ERR_FATAL est

utilisé.

bool **onCreate** (void)

Cette méthode est appelée durant l'instanciation du filtre. Si votre filtre alloue ou initialise d'autres ressources (comme des buffers), c'est le moment de le faire. Votre implémentation de cette méthode doit retourner FALSE en cas d'erreur et TRUE en cas de succès.

Lorsque votre filtre est instancié pour la première fois et que `yourfilter->onCreate()` est appelé, un nombre de propriétés est disponible comme montré dans la table ci-dessous.

Propriété	Contenu
FilterClass->filtername	Une chaîne contenant le nom du filtre est instanciée. Les filtres peuvent être enregistrés sous de noms multiples ainsi qu'avec des jokers. Utilisez cette propriété pour déterminer qu'elle est le nom utilisé.
FilterClass->params	Le contenu du paramètre passé params à la fonction <code>stream_filter_append</code> ou la fonction <code>stream_filter_prepend</code> .

void **onClose** (void)

Cette méthode est appelée durant l'extinction du filtre (généralement, lorsque le flux est fermé), et est exécutée **après** l'appel de la fonction `flush`. Si aucune ressource n'a été allouée ou créée durant `onCreate()`, c'est le moment de les libérer.

L'exemple ci-dessous implémente un filtre appelé `rot13`, sur le flux `foo-bar.txt`, qui réalise un chiffrement de type ROT-13 sur toutes les lettres lues ou écrites dans le flux.

Filtre sur les lettres majuscules sur le flux `foo-bar.txt`

```
<?php

/* Définition de la classe */
class strtoupper_filter extends php_user_filter {
    function filter($in, $out, &$consumed, $closing)
    {
        while ($bucket = stream_bucket_make_writeable($in)) {
            $bucket->data = strtoupper($bucket->data);
            $consumed += $bucket->datalen;
            stream_bucket_append($out, $bucket);
        }
        return PSFS_PASS_ON;
    }
}

/* Enregistrement de notre filtre avec PHP */
stream_filter_register("strtoupper", "strtoupper_filter")
    or die("Erreur lors de l'enregistrement du filtre");

$fp = fopen("foo-bar.txt", "w");

/* Attachement du filtre enregistré au flux que l'on vient d'ouvrir */
stream_filter_append($fp, "strtoupper");

fwrite($fp, "Ligne\n");
fwrite($fp, "Mot - 2\n");
fwrite($fp, "Facile comme 123\n");

fclose($fp);

/* Lecture du contenu
*/
readfile("foo-bar.txt");
```

```
?>
```

L'exemple ci-dessus va afficher :

```
LIGNE1
MOT - 2
FACILE COMME 123
```

Enregistrement d'une classe de filtre générique pour correspondre avec de multiples noms de filtres.

```
<?php
/* Définition de la classe*/
class string_filter extends php_user_filter {
    var $mode;

    function filter($in, $out, &$consumed, $closing)
    {
        while ($bucket = stream_bucket_make_writeable($in)) {
            if ($this->mode == 1) {
                $bucket->data = strtoupper($bucket->data);
            } elseif ($this->mode == 0) {
                $bucket->data = strtolower($bucket->data);
            }

            $consumed += $bucket->datalen;
            stream_bucket_append($out, $bucket);
        }
        return PSFS_PASS_ON;
    }

    function onCreate()
    {
        if ($this->filtername == 'str.toupper') {
            $this->mode = 1;
        } elseif ($this->filtername == 'str.tolower') {
            $this->mode = 0;
        } else {
            /* Quelques autres filtres str.* sont demandés,
               traitement de l'erreur avec PHP */
            return false;
        }

        return true;
    }
}

/* Enregistrement de notre filtre avec PHP */
stream_filter_register("str.*", "string_filter")
    or die("Failed to register filter");

$fp = fopen("foo-bar.txt", "w");

/* Attachement du filtre enregistré au flux que l'on vient d'ouvrir
   Nous pouvons alternativement passer à str.tolower ici */
stream_filter_append($fp, "str.toupper");

fwrite($fp, "Ligne1\n");
fwrite($fp, "Mot - 2\n");
fwrite($fp, "Facile comme 123\n");

fclose($fp);
```

```

/* Lecture du contenu
*/
readfile("foo-bar.txt");

?>

```

L'exemple ci-dessus va afficher :

```

LINE1
MOT - 2
FACILE COMME 123

```

Voir aussi [stream_wrapper_register](#) , [stream_filter_prepend](#) et [stream_filter_append](#) .

8.126.22 stream_filter_remove() : Supprime un filtre d'un flux

bool **stream_filter_remove** (resource stream_filter)

[stream_filter_append](#) supprime le dernier filtre ajouté au flux `stream_filter` avec [stream_filter_prepend](#) ou [stream_filter_append](#) . Toutes les données qui sont dans les buffers seront transmises au filtre, avant de le supprimer.

Filtrage dynamique d'un flux

```

<?php
/* Ouvre un fichier de test pour lecture et écriture */
$fp = fopen("test.txt", "rw");

$rot13_filter = stream_filter_append($fp, "string.rot13", STREAM_FILTER_WRITE);
fwrite($fp, "Ceci est ");
stream_filter_remove($rot13_filter);
fwrite($fp, "un test\n");

rewind($fp);
fpassthru($fp);
fclose($fp);

/* Affichage prévu
-----
Prpv rfg un test

*/
?>

```

Voir aussi [stream_filter_register](#) , [stream_filter_append](#) et [stream_filter_prepend](#) .

8.126.23 stream_get_contents() : Lit le reste d'un flux dans une chaîne

string **stream_get_contents** (resource handle , int maxlength , int offset)

Identique à [file_get_contents](#) , sauf que [stream_get_contents](#) opère sur un pointeur de fichier déjà ouvert et retourne le contenu restant, allant jusqu'à `maxlength` octets, dans une chaîne et commençant à la position `offset` .

8.126.24 `stream_get_filters()` : Liste les filtres enregistrés

array `stream_get_filters` (void)

`stream_get_filters` retourne un tableau indexé contenant la liste des filtres de flux disponibles sur le système.

Exemple avec `stream_get_filters`

```
<?php
$streamlist = stream_get_filters();
print_r($streamlist);
?>
```

L'affichage sera semblable a ce qui suit. Note : il peut y avoir plus ou moins de filtres suivant votre version de PHP

```
Array (
[0] => string.rot13
[1] => string.toupper
[2] => string.tolower
[3] => string.base64
[4] => string.quoted-printable
)
```

Voir aussi [stream_filter_register](#) et [stream_get_wrappers](#) .

8.126.25 `stream_get_line()` : Lit une ligne dans un flux

string `stream_get_line` (resource handle , int length , *string ending*)

`stream_get_line` lit une ligne de taille maximale length dans le flux handle . La lecture se termine lorsque length ont été lus, ou que la chaîne ending est trouvée dans le flux (et qui **n'est pas** incluse dans la valeur retournée), ou la fin du fichier (le premier des trois).

Si une erreur survient, `stream_get_line` retourne FALSE .

`stream_get_line` est presque identique à `fgets` hormis le fait qu'elle autorise d'autres délimiteur de fin de ligne que `\n` , `\r` et `\r\n` , et **ne retourne pas** le délimiteur lui-même.

Voir aussi [fread](#) , [fgets](#) et [fgetc](#) .

8.126.26 `stream_get_meta_data()` : Lit les en-têtes et données méta des flux

array `stream_get_meta_data` (resource stream)

`stream_get_meta_data` retourne les informations disponibles sur le flux stream . Ce flux peut être n'importe quel flux créé par [fopen](#) , [fsockopen](#) et [pfsockopen](#) . Le tableau résultant peut contenir les éléments suivants :

- `timed_out` (booléen) - TRUE si le flux a atteint de délai d'expiration en attendant des données durant le dernier appel aux fonctions `fread` et `fgets` .
- `blocked` (booléen) - TRUE si le flux est en mode bloquant. Voir aussi `stream_set_blocking` .
- `eof` (booléen) - TRUE si le flux a atteint la fin du fichier. Notez que pour les sockets, cette valeur peut être TRUE même si `unread_bytes` est non nul. Pour déterminer s'il reste des données à lire, utilisez plutôt la fonction `feof` .
- `unread_bytes` (entier) - le nombre d'octets actuellement placés dans le buffer interne à PHP.

Note

Vous ne devriez pas utiliser cette valeur dans un script.

Les éléments suivants ont été ajoutés en PHP 4.3 :

- `stream_type` (chaîne de caractères) - un nom, qui décrit l'implémentation sous-jacente de flux.
- `wrapper_type` (chaîne de caractères) - un nom qui décrit le gestionnaire de protocole pour ce flux. Voyez `Liste des protocoles supportés` pour plus d'informations sur les gestionnaires.
- `wrapper_data` (mixed) - des données spécifiques au gestionnaire liés à ce flux. Voyez `Liste des protocoles supportés` pour plus d'informations sur les gestionnaires et leurs données.
- `filters` (tableau) - un tableau contenant les noms de tous les filtres qui ont été attachés à ce flux. La documentation sur les filtres peut être trouvée sur l' `annexe concernant les filtres` .

Note

Cette fonction a été introduite en PHP 4.3, mais avant cette version, `socket_get_status` était utilisée pour lire les 4 premiers éléments, **uniquement pour les sockets** .

En PHP 4.3 et plus récent, `socket_get_status` est un alias de la fonction de cette fonction.

Note

Cette fonction n'est pas utilisable avec les sockets créées par les fonctions de l'extension `socket` .

Les éléments suivants ont été ajoutés en PHP 5.0:

- `mode` (chaîne de caractères) - le type d'accès requis pour ce flux (voir le tableau 1 de la référence de la fonction `fopen()`)
- `seekable` (booléen) - si on peut rechercher dans le flux courant.
- `uri` (chaîne de caractères) - l'URI/nom de fichier associé à ce flux.

8.126.27 `stream_get_transports()` : Liste les gestionnaires de transports de sockets disponibles

array `stream_get_transports` (void)

`stream_get_transports` retourne un tableau indexé contenant les noms des transports de sockets disponibles pour le système.

Exemple avec `stream_get_transports`

```
<?php
$portlist = stream_get_transports();
print_r($portlist);
?>
```

L'affichage sera identique au suivant. Note : il peut y avoir plus ou moins de transports dans votre version de PHP.

```
Array (
[0] => tcp
[1] => udp
[2] => unix
[3] => udg
)
?>
```

Voir aussi [stream_get_filters](#) et [stream_get_wrappers](#) .

8.126.28 stream_get_wrappers() : Liste les gestionnaires de flux

array **stream_get_wrappers** (void)

stream_get_wrappers retourne un tableau indexé contenant le nom de tous les gestionnaires de flux disponibles sur le système.

Exemple avec [stream_get_wrappers](#)

```
<?php
print_r(stream_get_wrappers());
?>
```

L'exemple ci-dessus va afficher quelque chose de similaire à :

```
Array
(
[0] => php
[1] => file
[2] => http
[3] => ftp
[4] => compress.bzip2
[5] => compress.zlib
)
```

Vérification de l'existence d'un gestionnaire de flux

```
<?php
// Vérification de l'existence d'un gestionnaire de flux bzip2
if (in_array('compress.bzip2', stream_get_wrappers())) {
    echo 'compress.bzip2:// support actif.';
} else {
    echo 'compress.bzip2:// support inactif.';
}
?>
```

Voir aussi [stream_wrapper_register](#) .

8.126.29 stream_register_wrapper() : Alias de stream_wrapper_register

Cette fonction est un alias de stream_wrapper_register . Elle est incluse pour assurer la compatibilité avec PHP 4.3.0 et PHP 4.3.1 uniquement. stream_wrapper_register doit être utilisée à la place.

8.126.30 stream_select() : Retourne l'équivalent de l'appel système select() sur un tableau de flux avec un délai d'expiration spécifié par tv_sec et tv_usec

int **stream_select** (array read , array write , array except , int tv_sec , int tv_usec)

stream_select accepte un tableau de flux et attend que l'un d'entre eux change de statut. Cette opération est équivalente à ce que fait la fonction socket_select , hormis le fait qu'elle travaille sur un flux.

Les flux qui sont listés dans le paramètre read seront surveillés en lecture, c'est-à-dire si de nouveaux octets sont disponibles en lecture (pour être précis, si une lecture ne bloquera pas, ce qui inclut aussi les flux qui sont en fin de fichier, auquel cas un appel à la fonction fread retournera une chaîne de taille 0).

Les flux qui sont listés dans le paramètre write seront surveillés en écriture (pour être précis, si une écriture ne bloquera pas).

Les flux qui sont listés dans le paramètre except seront surveillés pour voir si une exception est levée.

Note

Lorsque stream_select se termine, les tableaux read , write et except sont modifiés pour indiquer quel flux ont actuellement changé de statut.

Les paramètres tv_sec et tv_usec forment le **délai d'expiration** , tv_sec spécifie le nombre de secondes tandis que tv_usec , le nombre de microsecondes. Le paramètre timeout représente la limite supérieure du temps que stream_select doit attendre avant de se terminer. Si tv_sec et tv_usec sont tous les deux définis à 0 , stream_select n'attendra pas de données - à la place, elle se terminera immédiatement, indiquant le statut courant du flux. Si tv_sec vaut NULL , stream_select peut se bloquer indéfiniment, se terminant uniquement lorsqu'un événement sur un des flux surveillés survient (ou si un signal interrompt l'appel système).

En cas de succès, stream_select retourne le nombre de flux qui ont évolué, ce qui peut être 0 si le délai d'expiration a été atteint avant que les flux n'évoluent. En cas d'erreur, la fonction retournera FALSE et un avertissement sera renvoyé (cela peut apparaître si l'appel système est interrompu par un signal entrant).

Attention

Utiliser comme délai d'expiration une valeur de 0 vous permet de renvoyer instantanément le statut des flux, sinon, ce **n'est pas** une bonne idée d'utiliser une valeur de 0 dans une boucle, ce qui ferait que votre script consommerait beaucoup trop de temps processeur.

Il est préférable de spécifier une valeur de délai d'expiration de quelques secondes. Même si vous avez besoin d'exécuter d'autres codes en même temps, l'utilisation d'un délai d'expiration de moins de 200000 microsecondes devrait vous aider à réduire le temps processeur de votre script.

Souvenez-vous que le délai d'expiration représente le temps maximal qui s'écoulera ; stream_select se terminera dès que les flux désirés seront prêts à être utilisés.

Vous n'êtes pas obligé de passer tous les tableaux à la fonction stream_select . Vous pouvez en omettre certains, en utilisant un tableau vide ou la valeur NULL . N'oubliez pas que ces tableaux sont passés **par référence** et qu'ils seront modifiés lorsque stream_select s'achèvera.

Cet exemple surveille si des données arrivent pour être lues soit dans \$stream1 , soit dans \$stream2 . Si le délai d'expiration vaut 0 , la fonction se terminera immédiatement :

```
<?php
/* Preparation du tableau de flux lecture */
$read = array($stream1, $stream2);

if (false === ($num_changed_streams = stream_select($read, $write = NULL, $except = NULL, 0))) {
    /* Gestion d'erreur */
} elseif ($num_changed_streams > 0) {
    /* Au moins un des flux a évolué */
}
?>
```

Note

A cause d'une limitation du Moteur Zend actuel, il n'est pas possible de passer la valeur NULL directement comme paramètre d'une fonction qui s'attend à des paramètres passés par référence. Au lieu de cela, il est recommandé d'utiliser une variable temporaire, ou une expression dont le membre de gauche est une variable temporaire. Comme ceci :

```
<?php
stream_select($r, $w, $e = NULL, 0);
?>
```

Note

Assurez-vous de bien utiliser l'opérateur === lorsque vous recherchez des erreurs. Comme stream_select peut retourner 0, une comparaison effectuée à l'aide de == l'évaluerait à TRUE :

```
<?php
if (false === stream_select($r, $w, $e = NULL, 0)) {
    echo "stream_select() failed\n";
}
?>
```

Note

Si vous avez écrit ou lu dans un flux qui est retourné dans les tableaux de flux, soyez bien conscient que ces flux n'ont peut être pas écrit ou lu la totalité des données demandées. Soyez en mesure de lire un seul octet.

Note

Notes aux utilisateurs de Windows : stream_select utilisé sur une pipe retournée par proc_open peut causer la perte de données sous Windows 98.

Utiliser la fonction stream_select sur un descripteur de fichier retourné par proc_open échouera et retournera FALSE sous Windows.

Voir aussi stream_set_blocking .

8.126.31 `stream_set_blocking()` : Configure le mode bloquant d'un flux

bool `stream_set_blocking` (resource `stream` , int `mode`)

`stream_set_blocking` configure le mode bloquant du flux `stream` . Si `mode` vaut 0, `stream` sera configuré en mode non-bloquant, et s'il vaut 1, `stream` sera configuré en mode bloquant. Cet appel affecte les fonctions telles que `fgets` et `fread` qui lisent dans des flux. En mode non-bloquant, la fonction `fgets` s'exécute juste après son appel, alors qu'en mode bloquant, elle attendra des données.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

`stream_set_blocking` était appelée `set_socket_blocking` et plus tard `socket_set_blocking` mais ces appellations sont obsolètes.

Note

Avant PHP 4.3, cette fonction ne fonctionnait que sur les flux utilisant des sockets. Depuis PHP 4.3, cette fonction peut fonctionner avec n'importe quel flux qui supporte le mode non-bloquant (actuellement, les fichiers réguliers et les flux de sockets).

Voir aussi `stream_select` .

8.126.32 `stream_set_timeout()` : Configure la durée d'expiration d'un flux

bool `stream_set_timeout` (resource `stream` , int `seconds` , int `microseconds`)

`stream_set_timeout` configure la durée d'expiration du flux `stream` , exprimé comme la durée de seconds secondes et microseconds micro-secondes.

Lorsque le flux se termine, la clé 'timed_out' du tableau retourné par `stream_get_meta_data` est défini à TRUE , cependant, aucune erreur/alerte n'est générée.

Exemple avec `stream_set_timeout`

```
<?php
$fp = fsockopen("www.example.com", 80);
if (!$fp) {
    echo "Impossible d'ouvrir\n";
} else {

    fwrite($fp, "GET / HTTP/1.0\r\n\r\n");
    stream_set_timeout($fp, 2);
    $res = fread($fp, 2000);

    $info = stream_get_meta_data($fp);
    fclose($fp);

    if ($info['timed_out']) {
        echo 'Délai de connexion dépassé !';
    } else {
        echo $res;
    }
}
?>
```

Note

Depuis PHP 4.3, cette fonction peut (potentiellement) fonctionner avec n'importe quel flux. Avant PHP 4.3, les flux utilisant des sockets sont les seuls qui soient supportés dans le coeur de PHP, même si les autres extensions pourraient supporter cette fonction.

Note

Cette fonction ne fonctionne pas avec les opérations avancées comme [stream_socket_recvfrom](#), utilisez plutôt [stream_select](#) avec une durée d'expiration en paramètre.

Cette fonction était appelée auparavant [set_socket_timeout](#), et aussi [socket_set_timeout](#) mais ces appellations sont obsolètes.

Voir aussi [fsockopen](#) et [fopen](#).

8.126.33 [stream_set_write_buffer\(\)](#) : Configure la bufferisation de fichier pour un flux

int [stream_set_write_buffer](#) (resource stream , int buffer)

[fwrite](#) est habituellement configurée avec un buffer de 8 ko. Cela signifie que si deux processus veulent écrire dans le même flux de sortie (par exemple, un fichier), ils font une pause tous les 8 ko pour laisser l'autre écrire aussi. [stream_set_write_buffer](#) configure la taille du buffer des opérations d'écriture pour le flux stream avec la nouvelle valeur de buffer octets. Si buffer vaut 0, les opérations d'écriture ne seront pas bufferisées. Cela signifie que chaque appel à [fwrite](#) doit être terminé pour qu'un autre processus soit autorisé à écrire dans le fichier.

[stream_set_write_buffer](#) retourne 0 en cas de succès, et EOF si la requête échoue.

L'exemple ci-dessous illustre l'utilisation de [stream_set_write_buffer](#) pour créer un flux non bufferisé.

Exemple avec [stream_set_write_buffer](#)

```
$fp = fopen($file, "w");
if ($fp) {
    stream_set_write_buffer($fp, 0);
    fputs($fp, $output);
    fclose($fp);
}
```

Voir aussi [fopen](#) et [fwrite](#).

8.126.34 [stream_socket_accept\(\)](#) : Accepte une connexion sur une socket créée par [stream_socket_server](#)

resource [stream_socket_accept](#) (resource server_socket , float timeout , string peername)

[stream_socket_accept](#) accepte une connexion sur une socket créée précédemment avec [stream_socket_server](#). Si timeout est spécifié, le délai d'expiration de la socket sera remplacé par cette valeur, exprimée en secondes. L'adresse du client qui est connecté est retourné dans peername, si cette valeur est accessible pour ce type de transport.

peername peut aussi être déterminé plus tard avec [stream_socket_get_name](#).

Si [stream_socket_accept](#) échoue, elle retournera FALSE.

Attention

Cette fonction ne doit pas être utilisée avec un serveur de socket UDP. Il est recommandé d'utiliser à la place [stream_socket_recvfrom](#) et [stream_socket_sendto](#) .

Voir aussi [stream_socket_server](#) , [stream_socket_get_name](#) , [stream_set_blocking](#) , [stream_set_timeout](#) , [fgets](#) , [fgetss](#) , [fputs](#) , [fclose](#) , [feof](#) et l' [extension CURL](#) .

8.126.35 [stream_socket_client\(\)](#) : Ouvre une connexion socket Internet ou Unix

resource **[stream_socket_client](#)** (string *remote_socket* , int *errno* , string *errstr* , float *timeout* , int *flags* , resource *context*)

[stream_socket_client](#) initie un flux ou une connexion datagramme avec la destination *remote_socket* . Le type de socket créée est déterminé par le transport spécifié avec le formatage URL suivant : `transport://target` . Pour une socket Internet, (`AF_INET`) comme TCP et UDP, la cible de *remote_socket* sera une adresse IP ou un nom d'hôte. Pour une socket Unix, la cible doit être un fichier de socket du système.

Le paramètre optionnel *timeout* sert à imposer un délai d'expiration, en secondes, à l'appel système.

flags est un champ de bits qui peut être la combinaison de n'importe quelles options de connexion. Actuellement, les valeurs possibles pour ces options sont `STREAM_CLIENT_ASYNC_CONNECT` (défaut), `STREAM_CLIENT_ASYNC_CONNECT` et `STREAM_CLIENT_PERSISTENT` .

Note

Si vous n'avez pas besoin de mettre de délai d'expiration pour les opérations via la socket, utilisez la fonction [stream_set_timeout](#) , car *timeout* de [stream_socket_client](#) ne s'applique que lors de la connexion à la socket.

[stream_socket_client](#) retourne une ressource de flux, qui peut être utilisée avec d'autres fonctions de fichiers, comme [fgets](#) , [fgetss](#) , [fputs](#) , [fclose](#) et [feof](#)).

Si l'appel échoue, [stream_socket_client](#) va retourner `FALSE` et si les paramètres optionnels *errno* et *errstr* sont fournis, ils recevront l'erreur exacte qui est survenue dans le système lors de l'appel à `connect()` . Si la valeur retournée dans *errno* est 0 et que la fonction a retourné `FALSE` , c'est une indication que l'erreur est survenue avant l'appel à `connect()` . Cela est probablement dû à un problème d'initialisation de la socket. Notez que *errno* et *errstr* doivent être passés par référence.

Suivant votre environnement, les sockets Unix ou le délai d'expiration peuvent ne pas être disponibles. Une liste des transports disponibles sur le système est accessible via [stream_get_transports](#) . Voyez [Liste des modes de transports de sockets disponibles](#) pour une liste de transports natifs.

Le flux sera par défaut ouvert en mode bloquant. Vous pouvez le passer en mode non bloquant en utilisant la fonction [stream_set_blocking](#) .

Exemple avec [stream_socket_client](#)

```
<?php
$fp = stream_socket_client("tcp://www.example.com:80", $errno, $errstr, 30);
if (!$fp) {
    echo "$errstr ($errno)<br />\n";
} else {
    fputs($fp, "GET / HTTP/1.0\r\nHost: www.example.com\r\nAccept: */*\r\n\r\n");
}
```

```

while (!feof($fp)) {
    echo fgets ($fp,1024);
}
fclose($fp);
}
?>

```

L'exemple ci-dessous vous montre comment lire la date et l'heure sur un service UDP de type " daytime " (port 13) sur votre propre machine :

Utilisation de connexions UDP

```

<?php
$fp = stream_socket_client("udp://127.0.0.1:13", $errno, $errstr);
if (!$fp) {
    echo "ERREUR : $errno - $errstr<br />\n";
} else {
    fwrite($fp, "\n");
    echo fread($fp, 26);
    fclose($fp);
}
?>

```

Attention

Les sockets UDP sembleront parfois s'ouvrir sans erreur, même si l'hôte distant n'est pas accessible. Cette erreur ne deviendra apparente que lorsque vous essaierez de lire ou écrire des données avec cette socket. La raison à cela est que UDP est un protocole sans connexion, ce qui signifie que le système d'exploitation n'a pas à établir de lien pour la socket, jusqu'à ce qu'il commence à échanger des données.

Note

Lors de la spécification d'adresses IPv6 au format numérique (e.g. fe80::1) vous devez placer l'adresse IP entre crochets. Par exemple : tcp://[fe80::1]:80 .

Voir aussi [stream_socket_server](#) , [stream_set_blocking](#) , [stream_set_timeout](#) , [fgets](#) , [fgetss](#) , [fputs](#) , [fclose](#) , [feof](#) et l' [extension CURL](#) .

8.126.36 stream_socket_enable_crypto() : Active ou non le cryptage sur une socket déjà connectée

mixed **stream_socket_enable_crypto** (resource stream , bool enable , int crypto_type , resource session_stream)

Lorsqu'appelé avec le paramètre crypto_type , [stream_socket_enable_crypto](#) active le cryptage sur le flux stream en utilisant la méthode spécifiée.

Valeurs possibles pour le paramètre crypto_type

- STREAM_CRYPTO_METHOD_SSLv2_CLIENT
- STREAM_CRYPTO_METHOD_SSLv3_CLIENT
- STREAM_CRYPTO_METHOD_SSLv23_CLIENT
- STREAM_CRYPTO_METHOD_TLS_CLIENT
- STREAM_CRYPTO_METHOD_SSLv2_SERVER
- STREAM_CRYPTO_METHOD_SSLv3_SERVER
- STREAM_CRYPTO_METHOD_SSLv23_SERVER
- STREAM_CRYPTO_METHOD_TLS_SERVER

Une fois les paramètres de cryptage définis, le cryptage peut être activé et désactivé dynamiquement en passant TRUE ou FALSE dans le paramètre enable .

Si ce flux doit être défini avec les paramètres d'un flux actif crypté, passez la ressource de ce flux au paramètre `session_stream` .

Retourne TRUE en cas de succès, FALSE si la négociation a échoué ou 0 s'il n'y a pas assez de données et que vous devez essayer encore (uniquement pour les sockets non-bloquantes).

Exemple avec `stream_socket_enable_crypto`

```
<?php
$fp = stream_socket_client("tcp://monproto.example.com:31337", $errno, $errstr, 30);
if (!$fp) {
    die("Impossible de se connecter : $errstr ($errno)");
}

/* Activation du cryptage durant l'identification */
stream_socket_enable_crypto($fp, true, STREAM_CRYPTO_METHOD_SSLv23_CLIENT);
fwrite($fp, "USER Linus\r\n");
fwrite($fp, "PASS secret\r\n");

/* Désactivation du cryptage pour le reste */
stream_socket_enable_crypto($fp, false);

while ($motd = fgets($fp)) {
    echo $motd;
}
fclose($fp);
?>
```

[OpenSSL](#) et [Liste des modes de transports de sockets disponibles](#) .

8.126.37 `stream_socket_get_name()` : Lit le nom des sockets locale ou distante

string `stream_socket_get_name` (resource handle , bool want_peer)

`stream_socket_get_name` retourne le nom de la socket locale ou distante pour la connexion handle . Si `want_peer` vaut TRUE , le nom de la socket distante est retourné. S'il vaut FALSE , c'est celui de la socket locale.

Voir aussi [stream_socket_accept](#) .

8.126.38 `stream_socket_pair()` : Crée une paire de sockets connectées et indissociables

array `stream_socket_pair` (int domain , int type , int protocol)

`stream_socket_pair` crée une paire de sockets connectées et indissociables. Cette fonction est communément utilisée en IPC (InterProcess Communication) .

8.126.39 `stream_socket_recvfrom()` : Lit des données depuis une socket, connectée ou pas

string `stream_socket_recvfrom` (resource socket , int length , int flags , string address)

stream_socket_recvfrom accepte les données depuis une socket distante, jusqu'à un total de length octets. Si le paramètre address est fourni, il recevra l'adresse de la socket distante.

La valeur de flags peut être la combinaison des constantes suivantes :

STREAM_OOB	Traite les données en mode OOB (out-of-band).
STREAM_PEEK	Lit des données depuis la socket, mais n'utilise pas le buffer. Les prochains appels à <u>fread</u> ou <u>stream_socket_recvfrom</u> liront les mêmes données.
Exemple avec <u>stream_socket_recvfrom</u>	
<pre><?php /* Ouvre une socket sur le port 1234 de localhost */ \$server = stream_socket_server('tcp://127.0.0.1:1234'); /* Accepte une connexion */ \$socket = stream_socket_accept(\$server); /* Lit un paquet (1500 est la taille classique MTU) de données OOB */ echo "Received Out-Of-Band: '" . stream_socket_recvfrom(\$socket, 1500, STREAM_OOB) . "'\n"; /* Lit les données normales in-band, mais ne modifie rien */ echo "Data: '" . stream_socket_recvfrom(\$socket, 1500, STREAM_PEEK) . "'\n"; /* Relit le même paquet, mais vide le buffer. */ echo "Data: '" . stream_socket_recvfrom(\$socket, 1500) . "'\n"; /* Terminaison */ fclose(\$socket); fclose(\$server); ?></pre>	
Note	
Si le message reçu est plus grand que length , les données supplémentaires peuvent être détruites, suivant le type de socket utilisée (par exemple UDP).	

Voir aussi stream_socket_sendto , stream_socket_client et stream_socket_server .

8.126.40 stream_socket_sendto() : Envoie une message à la socket, connectée ou pas

int **stream_socket_sendto** (resource socket , string data , int flags , string address)

stream_socket_sendto envoie les données data à la socket socket . L'adresse de la socket est spécifiée lorsque la socket est créée, et sera utilisée si une autre adresse n'est pas spécifiée dans le paramètre address .

La valeur de flags peut être la combinaison des constantes suivantes :

STREAM_OOB	Traite les données en mode OOB (out-of-band).
Exemple avec <u>stream_socket_sendto</u>	
<pre><?php /* Ouvre une socket sur le port 1234 de localhost */ \$socket = stream_socket_client('tcp://127.0.0.1:1234'); /* Envoie des données directement */ fwrite(\$socket, "Normal data transmit."); /* Envoie d'autre données, en mode out of band. */ stream_socket_sendto(\$socket, "Mode out of Band.", STREAM_OOB);</pre>	

```

/* Fin */
fclose($socket);
?>

```

Voir aussi [stream_socket_recvfrom](#) , [stream_socket_client](#) et [stream_socket_server](#) .

8.126.41 stream_socket_server() : Crée une socket serveur Unix ou Internet

resource **stream_socket_server** (string local_socket , int errno , string errstr , int flags , resource context)

[stream_socket_server](#) crée un flux ou un datagramme sur la socket spécifiée local_socket . Le type de socket créée est déterminé par le transport spécifié avec le formatage URL suivant : transport://target . Pour une socket Internet, (AF_INET) comme TCP et UDP, la cible de remote_socket sera une adresse IP ou un nom d'hôte suivi de deux points et d'un numéro de port. Pour une socket Unix, la cible doit être un fichier de socket du système. flags est un champ de bits qui peut être la combinaison de n'importe quelle option de connexion. La valeur par défaut est STREAM_SERVER_BIND | STREAM_SERVER_LISTEN .

Note

Pour les sockets UDP, vous devez utiliser STREAM_SERVER_BIND comme valeur pour le paramètre flags .

[stream_socket_server](#) ne fait que créer une socket et, pour accepter des connexions, vous devez utiliser [stream_socket_accept](#) .

Si l'appel échoue, [stream_socket_server](#) va retourner FALSE et si les paramètres optionnels errno et errstr sont fournis, ils recevront l'erreur exacte qui est survenue dans le système lors des appels à connect() , bind() , et listen() . Si la valeur retournée dans errno est 0 et que la fonction a retourné FALSE , c'est une indication que l'erreur est survenue avant l'appel à bind() . Cela est probablement dû à un problème d'initialisation de la socket. Notez que errno et errstr doivent toujours être passés par référence.

Suivant votre environnement, les sockets Unix ou le délai d'expiration peuvent ne pas être disponibles. Une liste des transports disponibles sur le système est accessible via [stream_get_transports](#) . Voir [Liste des modes de transports de sockets disponibles](#) pour une liste complète des transports disponibles.

Exemple avec [stream_socket_server](#)

```

<?php
$socket = stream_socket_server("tcp://0.0.0.0:8000", $errno, $errstr);
if (!$socket) {
    echo "$errstr ($errno)<br />\n";
} else {
    while ($conn = stream_socket_accept($socket)) {
        fputs ($conn, 'L'heure locale est ' . date('n/j/Y g:i a') . "\n");
        fclose ($conn);
    }
    fclose($socket);
}
?>

```

L'exemple ci-dessous vous montre comment lire la date et l'heure sur un service UDP de type " daytime " (port 13) sur votre propre machine, tel que présenté avec la fonction [stream_socket_client](#)

:

Note

La plupart des systèmes ont besoin d'un accès administrateur pour ouvrir une socket au-delà des ports 1024.

Utiliser un serveur de socket UDP

```
<?php
$socket = stream_socket_server("udp://0.0.0.0:13", $errno, $errstr, STREAM_SERVER_BIND);
if (!$socket) {
    echo "ERROR: $errno - $errstr<br />\n";
} else {
    while ($conn = stream_socket_accept($socket)) {
        fwrite($conn, date("D M j H:i:s Y\r\n"));
        fclose($conn);
    }
    fclose($socket);
}
?>
```

Note

Lors de la spécification d'adresses IPv6 au format numérique (e.g. fe80::1) vous devez placer l'adresse IP entre crochets. Par exemple : tcp://[fe80::1]:80 .

Voir aussi [stream_socket_client](#) , [stream_set_blocking](#) , [stream_set_timeout](#) , [fgets](#) , [fgetss](#) , [fwrite](#) , [fclose](#) , [feof](#) et l' [extension CURL](#) .

8.126.42 [stream_wrapper_register\(\)](#) : Enregistre une enveloppe URL, implémentée comme une classe PHP

bool [stream_wrapper_register](#) (string protocol , string classname)

[stream_wrapper_register](#) permet d'implémenter des gestionnaires de protocole et de flux, à utiliser avec toutes les autres fonctions de fichiers, comme [fopen](#) , [fread](#) etc.

Pour implémenter une enveloppe, il faut définir une classe avec la liste des membres définie ci-dessous. Lorsque quelqu'un ouvre le flux, PHP va créer une instance de la classe classname et appeler les méthodes de cette instance. Il faut implémenter ces méthodes exactement comme décrit ci-dessous : sinon, les comportements seront indéfinis.

Note

Depuis PHP 5.0.0, l'instance de la classe classname contiendra une propriété context , faisant référence à la ressource de contexte qui peut être accédée avec la fonction [stream_context_get_options](#) . Si aucun contexte n'est passé à la fonction de création du flux, context sera défini à NULL .

[stream_wrapper_register](#) retourne FALSE si le protocole protocol a déjà un gestionnaire attribué.

bool [stream_open](#) (string path , string mode , int options , string opened_path)

Cette méthode est appelée immédiatement après la création de votre flux. path spécifie l'URL qui doit être passée à la fonction [fopen](#) et ce que cet objet est supposé y lire. Vous pouvez utiliser [parse_url](#) pour l'analyser.

mode est le mode d'ouverture du fichier, comme expliqué dans [fopen](#) . Vous êtes responsable de la vérification de la validité du paramètre mode avec le chemin path fourni.

8.126.41 [stream_socket_server\(\)](#) : Crée une socket serveur Unix ou Internet

1872

options contient des options supplémentaires, utilisées par les API de flux. Il peut contenir une ou plusieurs des options suivantes, combinées avec l'opérateur OR.

Option	Description
STREAM_USE_PATH	Si path est relatif, recherche la ressources en utilisant la configuration de l' <code>include_path</code> .
STREAM_REPORT_ERRORS	Si cette option est activée, la classe est responsable de la génération d'erreurs avec <code>trigger_error</code> durant l'ouverture du flux. Si cette option n'est pas activée, il ne faut générer aucune erreur.

Si le paramètre path est ouvert avec succès, et que STREAM_USE_PATH est activé dans le paramètre options , il faut affecter à `opened_path` le chemin complet de la ressource ou du fichier qui a été réellement ouvert.

Si la ressource demandée a été ouverte, il faut retourner TRUE , ou, sinon, FALSE

void **stream_close** (void)

Cette méthode est appelée lorsque le flux est fermé, grâce à la fonction `fclose` . Vous devez libérer toutes les ressources verrouillées ou réservées par le flux.

string **stream_read** (int count)

Cette méthode est appelée suite à l'utilisation des fonctions `fread` et `fgets` . Vous devez lire jusqu'à count octets de données à partir de la position courante d'écriture ou de lecture, sous la forme de chaîne de caractères. S'il y a moins que count octets disponibles, vous devez en retourner autant que possible. Si aucune autre donnée n'est disponible, FALSE sera retourné ou une chaîne vide. Vous devez aussi tenir à jour la position du pointeur d'écriture/lecture dans le flux, en ajoutant ou retranchant le nombre d'octets lus.

int **stream_write** (string data)

Cette méthode est appelée lorsque la fonction `fwrite` est utilisée. Vous devez stocker les données data dans le flux. S'il n'y a pas assez de place, essayez d'en stocker le maximum. Vous devriez aussi retourner le nombre d'octets que vous avez réussi à écrire, ou bien 0 si aucun n'a pu être stocké. Vous devez aussi tenir à jour la position du pointeur d'écriture/lecture dans le flux, en ajoutant ou retranchant le nombre d'octets écrits.

bool **stream_eof** (void)

Cette méthode est appelée lorsque la fonction `feof` est utilisée. Vous devez retourner TRUE si la position de lecture se situe à la fin du fichier et s'il n'y a plus de données disponibles pour la lecture, ou bien FALSE sinon.

int **stream_tell** (void)

Cette méthode est appelée lorsque la fonction `ftell` est utilisée. Vous devez retourner la position actuelle du pointeur de lecture/écriture dans le flux.

bool **stream_seek** (int offset , int whence)

Cette méthode est appelée lorsque la fonction `fseek` est utilisée. Vous devez modifier la position du pointeur de lecture/écriture en fonction des paramètres d'offset offset et de direction whence . Reportez-vous à la fonction `fseek` pour plus de détails sur ces paramètres. Retournez TRUE si la position a été modifiée, et FALSE sinon.

bool **stream_flush** (void)

Cette méthode est appelée lorsque la fonction flush est utilisée. Si vous avez mis des données dans un système de cache pour votre flux, mais qu'elles ne sont pas encore stockées de manière pérenne, c'est le moment de le faire. Retournez TRUE si les données cachées ont pu être stockées avec succès (il n'y a plus de données à stocker), ou bien FALSE si les données n'ont pu être stockées.

array **stream_stat** (void)

Cette méthode est appelée en réponse à la fonction fstat sur un flux, et retourne un tableau, avec les valeurs appropriées pour le flux.

bool **unlink** (string path)

Cette méthode est appelée en réponse à la fonction unlink avec une URL, et doit tenter de supprimer l'objet identifié par path . La fonction doit retourner TRUE en cas de succès, et FALSE en cas d'échec. Afin de retourner le bon message d'erreur, ne définissez pas cette méthode si votre gestionnaire ne le supporte pas.

Note

La méthode enveloppe d'utilisateur unlink n'est pas supportée avant PHP 5.0.0.
--

bool **rename** (string path_from , string path_to)

Cette méthode est appelée en réponse à la fonction rename avec une URL associée avec l'enveloppe et doit tenter de renommer l'objet identifié par path_from en path_to . Cela doit retourner TRUE en cas de succès ou FALSE en cas d'erreur. Afin de pouvoir retourner le message d'erreur approprié, ne définissez pas cette méthode si votre enveloppe ne supporte pas le renommage.

Note

La méthode enveloppe d'utilisateur rename n'est pas supportée avant PHP 5.0.0.
--

bool **mkdir** (string path , int mode , int options)

Cette méthode est appelée en réponse à la fonction mkdir avec une URL associée avec l'enveloppe et doit tenter de créer le répertoire spécifié par path . Cela doit retourner TRUE en cas de succès ou FALSE en cas d'erreur. Afin de pouvoir retourner le message d'erreur approprié, ne définissez pas cette méthode si votre enveloppe ne supporte pas la création de répertoire. Les valeurs possibles pour le paramètre options sont STREAM_REPORT_ERRORS et STREAM_MKDIR_RECURSIVE .

Note

La méthode enveloppe d'utilisateur mkdir n'est pas supportée avant PHP 5.0.0.

bool **rmdir** (string path , int options)

Cette méthode est appelée en réponse à la fonction rmdir avec une URL associée avec l'enveloppe et doit tenter de supprimer le répertoire spécifié par path . Cela doit retourner TRUE en cas de succès ou FALSE en cas d'erreur. Afin de pouvoir retourner le message d'erreur approprié, ne définissez pas cette méthode si votre enveloppe ne supporte pas la suppression de répertoire. La valeur possible pour le paramètre options est STREAM_REPORT_ERRORS .

Note

La méthode enveloppe d'utilisateur rmdir n'est pas supportée avant PHP 5.0.0.

bool **dir_opendir** (string path , int options)

Cette méthode est appelée immédiatement lorsque votre flux est créé, pour examiner le contenu d'un dossier avec `opendir` . `path` spécifie l'URL qui est passée à `opendir` et que cet objet doit explorer. Vous pouvez utiliser `parse_url` pour la scinder en morceaux.

array **url_stat** (string `path` , int `flags`)

Cette méthode est appelée en réponse à la fonction `stat` avec une URL associée avec l'enveloppe et doit retourner autant d'éléments en commun avec la fonction système que possible. Les valeurs non connues ou non disponibles doivent être définies en une valeur rationnelle (habituellement, 0).

`flags` représente les options additionnelles définies par l'API. Il peut prendre une ou plusieurs des valeurs suivantes séparées par OR.

Option	Description
STREAM_URL_STAT_LINK	Pour les ressources avec la possibilité de relier d'autres ressources (comme par exemple HTTP Location: forward ou un lien symbolique d'un système de fichiers). Cette option spécifie que seul les informations à propos du lien lui-même peuvent être retournées, et non pas les ressources pointées par le lien. Cette option est définie en réponse à l'appel à la fonction <code>lstat</code> , <code>is_link</code> , ou <code>filetype</code> .
STREAM_URL_STAT_QUIET	Si cette option est définie, votre enveloppe ne devra contenir aucune erreur. Si cette option n'est pas définie, vous êtes responsable du traitement des erreurs en utilisant la fonction <code>trigger_error</code> pendant l'évaluation du chemin.

string **dir_readdir** (void)

Cette méthode est appelée en réponse à la fonction `readdir` et doit retourner une chaîne représentant le prochain fichier ouvert par `dir_opendir` .

bool **dir_rewinddir** (void)

Cette méthode est appelée en réponse à `rewinddir` et doit remettre à zéro les résultats de `dir_readdir` . i.e. : le prochain appel à `dir_readdir` doit retourner la première ligne située dans le dossier ouvert par `dir_opendir` .

bool **dir_closedir** (void)

Cette méthode est appelée en réponse à `closedir` . Vous devez libérer toutes les ressources qui ont été réservées durant l'ouverture et l'utilisation du dossier.

L'exemple ci-dessous implémente un gestionnaire de protocole pour le protocole `var://`, qui permet l'écriture et la lecture de variables globales en utilisant un flux de fichier standard, et les fonctions classiques telles que `fread` . Le protocole `var://` implémenté ci-dessous, étant donné l'URL `var://foo` va écrire ou lire les données dans `$GLOBALS["foo"]` .

Une classe de flux pour accéder aux variables globales
<pre> <?php class VariableStream { var \$position; var \$varname; function stream_open(\$path, \$mode, \$options, &\$opened_path) { \$url = parse_url(\$path); } </pre>

```

        $this->varname = $url["host"];
        $this->position = 0;

        return true;
    }

    function stream_read($count)
    {
        $ret = substr($GLOBALS[$this->varname], $this->position, $count);
        $this->position += strlen($ret);
        return $ret;
    }

    function stream_write($data)
    {
        $left = substr($GLOBALS[$this->varname], 0, $this->position);
        $right = substr($GLOBALS[$this->varname], $this->position + strlen($data));
        $GLOBALS[$this->varname] = $left . $data . $right;
        $this->position += strlen($data);
        return strlen($data);
    }

    function stream_tell()
    {
        return $this->position;
    }

    function stream_eof()
    {
        return $this->position >= strlen($GLOBALS[$this->varname]);
    }

    function stream_seek($offset, $whence)
    {
        switch($whence) {
            case SEEK_SET:
                if ($offset < strlen($GLOBALS[$this->varname]) && $offset >= 0) {
                    $this->position = $offset;
                    return true;
                } else {
                    return false;
                }
                break;

            case SEEK_CUR:
                if ($offset >= 0) {
                    $this->position += $offset;
                    return true;
                } else {
                    return false;
                }
                break;

            case SEEK_END:
                if (strlen($GLOBALS[$this->varname]) + $offset >= 0) {
                    $this->position = strlen($GLOBALS[$this->varname]) + $offset;
                    return true;
                } else {
                    return false;
                }
                break;

            default:
                return false;
        }
    }
}

```

```

stream_wrapper_register("var", "VariableStream")
    or die("Failed to register protocol");

$myvar = "";

$fp = fopen("var://myvar", "r+");

fwrite($fp, "line1\n");
fwrite($fp, "line2\n");
fwrite($fp, "line3\n");

rewind($fp);
while(!feof($fp)) {
    echo fgets($fp);
}
fclose($fp);
var_dump($myvar);

?>

```

8.126.43 `stream_wrapper_restore()` : Restaure un gestionnaire d'URL supprimé

bool `stream_wrapper_restore` (string protocol)

`stream_wrapper_restore` restaure le gestionnaire protocol précédemment supprimé avec `stream_wrapper_unregister` .

8.126.44 `stream_wrapper_unregister()` : Supprime un gestionnaire d'URL

bool `stream_wrapper_unregister` (string protocol)

`stream_wrapper_unregister` supprime le gestionnaire d'URL avec le protocole protocol . Une fois que le gestionnaire est retiré, vous pouvez installer le vôtre, avec la fonction `stream_wrapper_register` ou le réactiver plus tard avec `stream_wrapper_restore` .

8.127 Chaînes de caractères (Strings)

8.127.1 Introduction

Ces fonctions permettent de manipuler des chaînes de caractères de différentes façons. Certaines fonctionnalités plus spécialisées sont disponibles dans les sections d'expressions rationnelles et de [gestion des URL](#) .

Pour plus de détails sur le comportement des chaînes de caractères, notamment avec les guillemets simples et doubles, les séquences d'échappement, reportez-vous à la section [Chaînes de caractères](#) dans la section [Types](#) .

8.127.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.127.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.127.4 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

CRYPT_SALT_LENGTH [entier](#)
CRYPT_STD_DES [entier](#)
CRYPT_EXT_DES [entier](#)
CRYPT_MD5 [entier](#)
CRYPT_BLOWFISH [entier](#)
HTML_SPECIALCHARS ([entier](#))
HTML_ENTITIES ([entier](#))
ENT_COMPAT ([entier](#))
ENT_QUOTES ([entier](#))
ENT_NOQUOTES ([entier](#))
CHAR_MAX ([entier](#))
LC_CTYPE ([entier](#))
LC_NUMERIC ([entier](#))
LC_TIME ([entier](#))
LC_COLLATE ([entier](#))
LC_MONETARY ([entier](#))
LC_ALL ([entier](#))
LC_MESSAGES ([entier](#))
STR_PAD_LEFT ([entier](#))
STR_PAD_RIGHT ([entier](#))
STR_PAD_BOTH ([entier](#))

8.127.5 Voir aussi

Pour des fonctions encore plus puissantes de gestion et manipulation des chaînes, reportez-vous aux [expressions rationnelles POSIX](#) et [expressions rationnelles Perl](#) .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Constantes pré-définies](#)
- [Voir aussi](#)
- [addslashes](#)
- [addslashes](#)
- [bin2hex](#)
- [chop](#)
- [chr](#)
- [chunk_split](#)
- [convert_cyr_string](#)
- [convert_uudecode](#)
- [convert_uuencode](#)
- [count_chars](#)
- [crc32](#)
- [crypt](#)
- [echo](#)
- [explode](#)
- [fprintf](#)
- [get_html_translation_table](#)
- [hebrew](#)
- [hebrevc](#)
- [html_entity_decode](#)
- [htmlentities](#)
- [htmlspecialchars_decode](#)
- [htmlspecialchars](#)
- [implode](#)
- [join](#)
- [levenshtein](#)
- [localeconv](#)
- [ltrim](#)
- [md5_file](#)
- [md5](#)
- [metaphone](#)
- [money_format](#)
- [nl_langinfo](#)
- [nl2br](#)
- [number_format](#)
- [ord](#)
- [parse_str](#)
- [print](#)
- [printf](#)
- [quoted_printable_decode](#)
- [quotemeta](#)
- [rtrim](#)
- [setlocale](#)
- [sha1_file](#)
- [sha1](#)

- [similar_text](#)
- [soundex](#)
- [sprintf](#)
- [sscanf](#)
- [str_ireplace](#)
- [str_pad](#)
- [str_repeat](#)
- [str_replace](#)
- [str_rot13](#)
- [str_shuffle](#)
- [str_split](#)
- [str_word_count](#)
- [strcasecmp](#)
- [strchr](#)
- [strcmp](#)
- [strcoll](#)
- [strcspn](#)
- [strip_tags](#)
- [stripslashes](#)
- [stripos](#)
- [stripslashes](#)
- [stristr](#)
- [strlen](#)
- [strnatcasecmp](#)
- [strnatcmp](#)
- [strncasecmp](#)
- [strncmp](#)
- [strpbrk](#)
- [strpos](#)
- [strrchr](#)
- [strev](#)
- [stripos](#)
- [strrpos](#)
- [strspn](#)
- [strstr](#)
- [strtok](#)
- [strtolower](#)
- [strtoupper](#)
- [strtr](#)
- [substr_compare](#)
- [substr_count](#)
- [substr_replace](#)
- [substr](#)
- [trim](#)
- [ucfirst](#)
- [ucwords](#)
- [fprintf](#)
- [vprintf](#)
- [vsprintf](#)
- [wordwrap](#)

8.127.7 addslashes() : Ajoute des anti-slash dans une chaîne

string **addslashes** (string str)

addslashes retourne la chaîne str , après avoir échappé tous les caractères qui doivent l'être, pour être utilisée dans une requête de base de données. Ces caractères sont les guillemets simples ('), guillemets doubles ("), anti-slash (\) et NUL (le caractère NULL).

Un exemple d'utilisation d' addslashes est lorsque vous entrez des données dans une base de données. Par exemple, pour insérer le nom O'reilly dans la base, vous aurez besoin de le protéger. La plupart des base de données le font avec \ ce qui donnerait O\'reilly . Cela ne servirait qu'à insérer le nom dans la base de données, le \ ajouté ne sera pas inséré. Si la directive PHP magic_quotes_sybase est activée, ' sera protégé par un autre ' .

La directive PHP magic_quotes_gpc est à on par défaut, et elle appelle addslashes sur toutes les données GET, POST et COOKIE. N'utilisez pas addslashes sur des données déjà protégées avec magic_quotes_gpc sinon vous doublerez les protections. La fonction get_magic_quotes_gpc est utile pour vérifier ce paramètre.

Exemple avec addslashes

```
<?php
$str = "Votre nom est-il O'reilly ?";
echo addslashes($str); // Votre nom est-il O\'reilly ?
?>
```

Voir aussi stripslashes , htmlspecialchars , quotemeta et get_magic_quotes_gpc .

8.127.8 bin2hex() : Convertit des données binaires en représentation hexadécimale

string **bin2hex** (string str)

bin2hex retourne la chaîne str dont tous les caractères sont représentés par leur équivalent hexadécimal. La chaîne retournée est une chaîne ASCII. La conversion supporte les caractères binaires, et utilise les bits de poids forts en premier.

Voir aussi pack et unpack .

8.127.9 chop() : Alias de rtrim

Cette fonction est un alias de : rtrim .

Note

chop est différente de son équivalent Perl chop() , qui supprime le dernier caractère dans la chaîne.

8.127.10 chr() : Retourne un caractère spécifique

string **chr** (int ascii)

chr retourne une chaîne d'un seul caractère, dont le code ASCII est donné par le paramètre `ascii` .

Exemple avec `chr`

```
<?php
// Ajoute un caractère d'échappement à la fin de la chaîne $str
$str .= chr(27);

// Ceci est souvent plus pratique, et réalise la même chose
$str = sprintf("La chaîne se finit par un caractère d'échappement : %c", 27);

?>
```

Vous pouvez trouver une table des codes ASCII ici : <http://www.asciitable.com> .

Cette fonction est l'inverse de `ord` . Voir aussi `sprintf` , et le caractère de formatage `%c` .

8.127.11 `chunk_split()` : Scinde une chaîne

string `chunk_split` (string `body` , int `chunklen` , string `end`)

`chunk_split` scinde la chaîne `body` en segments de `chunklen` octets de longueur. Cette fonction est très pratique pour convertir les résultats de `base64_encode` au format de la RFC 2045. Elle insère le paramètre `end` (par défaut, `"\r\n"`) tous les `chunklen` caractères (par défaut, 76). Elle retourne la nouvelle chaîne, et ne modifie pas la chaîne originale.

Exemple avec `chunk_split`

```
<?php
// Formater des données pour suivre la norme RFC 2045
$new_string = chunk_split(base64_encode($data));

?>
```

Voir aussi `explode` , `split` , `wordwrap` et `RFC 2045` .

8.127.12 `convert_cyr_string()` : Convertit une chaîne d'un jeu de caractères cyrillique à l'autre

string `convert_cyr_string` (string `str` , string `from` , string `to`)

`convert_cyr_string` retourne la chaîne `str` , après l'avoir traduite depuis le jeu de caractères cyrillique `from` vers `to` . Les paramètres `from` et `to` sont des caractères qui représentent les jeux de caractères d'origine et de destination. Les types suivants sont supportés :

- `k` : koi8-r
- `w` : windows-1251
- `i` : iso8859-5
- `a` : x-cp866
- `d` : x-cp866
- `m` : x-mac-cyrillic

Note

Cette fonction gère les chaînes binaires.

8.127.13 `convert_uuencode()` : Décode une chaîne au format uuencode

string `convert_uuencode` (string data)

`convert_uuencode` décode une chaîne au format uuencode.

Exemple avec `convert_uuencode`

```
<?php
/* Pouvez-vous imaginer ce que cela va afficher ? :) */
echo convert_uuencode("+22!L;W9E(%!(4\"$`\\n`");
?>
```

Voir aussi `convert_uuencode` .

8.127.14 `convert_uuencode()` : Encode une chaîne de caractères en utilisant l'algorithme uuencode

string `convert_uuencode` (string data)

`convert_uuencode` encode une chaîne de caractères en utilisant l'algorithme uuencode.

Uuencode traduit toutes les chaînes (y compris les binaires) en caractères imprimables, pour assurer leur transmission sur Internet. Les données au format uuencode sont environ 35 % plus grandes que les originales.

Exemple avec `convert_uuencode`

```
<?php
$some_string = "test\ntexte texte\r\n";

echo convert_uuencode($some_string);
?>
```

Voir aussi `convert_uuencode` et `base64_encode` .

8.127.15 `count_chars()` : Retourne des statistiques sur les caractères utilisés dans une chaîne

mixed `count_chars` (string string , int mode)

`count_chars` compte le nombre d'occurrences de tous les octets présents dans la chaîne string et retourne différentes statistiques. Le paramètre optionnel mode vaut par défaut 0. Suivant la valeur de mode , `count_chars` retourne les informations suivantes :

- 0 - un tableau avec l'octet en index, et la fréquence correspondante pour chaque octet.
- 1 - identique à 0 mais seules les fréquences supérieures à zéro sont listées.
- 2 - identique à 0 mais seules les fréquences nulles sont listées.
- 3 - une chaîne contenant tous les octets utilisés est retournée.
- 4 - une chaîne contenant tous les octets non utilisés est retournée.

Exemple avec `count_chars`

```
<?php
$data = "Deux D et un F.";

foreach (count_chars($data, 1) as $i => $val) {
    echo "Il y avait $val occurrence(s) de \"", chr($i) , "\" dans la phrase.\n";
}
?>
```

L'exemple ci-dessus va afficher :

```
Il y avait 4 occurrence(s) de " " dans la phrase.
Il y avait 1 occurrence(s) de "." dans la phrase.
Il y avait 2 occurrence(s) de "D" dans la phrase.
Il y avait 1 occurrence(s) de "F" dans la phrase.
Il y avait 2 occurrence(s) de "e" dans la phrase.
Il y avait 1 occurrence(s) de "n" dans la phrase.
Il y avait 1 occurrence(s) de "t" dans la phrase.
Il y avait 2 occurrence(s) de "u" dans la phrase.
Il y avait 1 occurrence(s) de "x" dans la phrase.
```

Voir aussi [strpos](#) et [substr_count](#) .

8.127.16 `crc32()` : Calcule la somme de contrôle CRC32

int **crc32** (string str)

`crc32` génère la somme de contrôle cyclique CRC32, calculée sur 32 bits, et appliquée à la chaîne str . Cette fonction est généralement utilisée pour valider l'intégrité de données durant une transmission.

Comme le type d'entier de PHP est signé, et que de nombreuses sommes de contrôle CRC32 aboutissent à un nombre négatif, vous devez utiliser le format "%u" de la fonction [sprintf](#) ou [printf](#) pour obtenir la représentation littérale de la somme de contrôle CRC32.

Cet exemple illustre comment afficher la somme de contrôle avec la fonction [printf](#) :

Afficher une somme de contrôle CRC32

```
<?php
$checksum = crc32("Le vif zéphyr jubile sur les kumquats du clown gracieux.");
printf("%u\n", $checksum);
?>
```

Voir aussi [md5](#) et [sha1](#) .

8.127.17 `crypt()` : Chiffage indéchiffrable (hashing)

string **crypt** (string str , string salt)

`crypt` retourne la chaîne str chiffrée avec l'algorithme standard Unix *DES* , ou bien un des algorithmes disponibles sur la machine. L'argument optionnel salt sera utilisé comme base pour le chiffrement. Reportez-vous aux pages de manuel Unix pour plus de détails sur la fonction crypt.

8.127.15 `count_chars()` : Retourne des statistiques sur les caractères utilisés dans une chaîne 1884

Si l'argument `salt` n'est pas fourni, PHP en générera un lui-même, à chaque appel à cette fonction.

Certains systèmes supportent plus d'un type de chiffrement. En fait, il arrive que le chiffrement DES standard soit remplacé par un algorithme de chiffrement MD5. Le choix du type de chiffrement est effectué en se basant sur la valeur du `salt`. A l'installation, PHP détermine les possibilités de la fonction `crypt`, et acceptera des `salt` pour d'autres types de chiffrements. Si aucun `salt` n'est fourni, PHP va en générer un de deux caractères, à moins que le système par défaut soit MD5, auquel cas un `salt` compatible MD5 sera généré. PHP définit une constante appelée `CRYPT_SALT_LENGTH` qui vous dira si un `salt` de deux caractères s'applique à votre système, ou bien si c'est un `salt` de 12 caractères.

Si vous utilisez le `salt` généré, pensez bien que le `salt` est généré une seule fois. Si vous appelez cette fonction plusieurs fois, cela peut avoir des impacts sur la sécurité et l'apparence du résultat.

`crypt`, lorsqu'elle est utilisée avec le chiffrement standard DES, retourne le `salt` dans les deux premiers caractères de la chaîne retournée. Elle n'utilise que les 8 premiers caractères de `str`, ce qui fait que toutes les chaînes plus longues, qui ont les mêmes premiers 8 octets retourneront le même résultat (tant que le `salt` est toujours le même).

Sur les systèmes où `crypt` supporte plusieurs types de chiffrements, les constantes suivantes sont mises à 0 ou 1, suivant que le type correspondant est disponible :

- `CRYPT_STD_DES` : chiffrement DES standard à 2 caractères
- `CRYPT_EXT_DES` : chiffrement DES étendu à 9 caractères
- `CRYPT_MD5` : chiffrement MD5 à 12 caractères commençant par \$1\$
- `CRYPT_BLOWFISH` : chiffrement Blowfish à 16 caractères commençant par \$2\$ ou \$2a\$.

Note
Il n'existe pas de fonction de déchiffrement, car la fonction <code>crypt</code> utilise un algorithme à un seul sens (injection).
Exemple avec <code>crypt</code>
<pre><?php // laissons le salt initialisé par PHP \$mot_de_passe = crypt("monmotdepasse"); /* Il vaut mieux passer le résultat complet de crypt() comme salt nécessaire pour le chiffrement du mot de passe, pour éviter les problèmes entre les algorithmes utilisés (comme nous le disons ci-dessus, le chiffrement standard DES utilise un salt de 2 caractères, mais un chiffrement MD5 utilise un salt de 12). */ if (crypt(\$utilisateur_input,\$mot_de_passe_input) == \$mot_de_passe) { echo "Mot de passe correct!"; } ?></pre>
Utilisation de <code>crypt</code> avec <code>htpasswd</code>
<pre><?php // Définition du mot de passe \$password = 'monmotdepasse'; // Récupération du hash, on laisse le salt se générer automatiquement \$hash = crypt(\$password); ?></pre>
Utilisation de <code>crypt</code> avec différents types de cryptage

```

<?php
if (CRYPT_STD_DES == 1) {
    echo 'DES standard : ' . crypt('rasmuslerdorf', 'rl') . "\n";
}

if (CRYPT_EXT_DES == 1) {
    echo 'DES étendu : ' . crypt('rasmuslerdorf', '_J9..rasm') . "\n";
}

if (CRYPT_MD5 == 1) {
    echo 'MD5 :          ' . crypt('rasmuslerdorf', '$1$rasmusle$') . "\n";
}

if (CRYPT_BLOWFISH == 1) {
    echo 'Blowfish :     ' . crypt('rasmuslerdorf', '$2a$07$rasmuslerd.....$') . "\n";
}
?>

```

L'exemple ci-dessus va afficher :

```

DES standard : rl.3StKT.4T8M
DES étendu : _J9..rasmBYk8r9AiWNc
MD5 : $1$rasmusle$rISCgZzpwk3UhDidwXvin0
Blowfish : $2a$07$rasmuslerd.....nldrcHdxcUxWomQX9j6kvERCFjTg7Ra

```

Voir aussi [md5](#) et l'[extension Mcrypt](#) .

8.127.18 echo() : Affiche une chaîne de caractères

void **echo** (string arg1 , string ...)

Affiche tous les paramètres.

echo n'est pas vraiment une fonction (c'est techniquement une structure du langage), cela fait que vous n'êtes pas obligé d'utiliser des parenthèses. echo (contrairement à d'autres structures de langage) ne se comporte pas comme une fonction, il ne peut donc pas être utilisé dans le contexte d'une fonction. De même, si vous voulez passer plusieurs paramètres à echo , les paramètres ne doivent pas être entourés de parenthèses.

Exemples avec echo

```

<?php
echo "Bonjour le monde";

echo "Cet echo() se
répartit sur plusieurs lignes. Il affiche aussi les
nouvelles lignes";

echo "Cet echo() se\nrépartit sur plusieurs lignes. Il affiche aussi les\nnouvelles lignes";

echo "L'échappement de caractères se fait : \"comme ceci\".";

// Vous pouvez utiliser des variables avec echo()
$foo = "foobar";
$bar = "barbaz";

echo "foo vaut \"\$foo\""; // foo vaut "foobar"

// Vous pouvez aussi utiliser des tableaux

```

```

$bar = array("clé" => "foo");

echo "c'est {$bar['clé']} !"; // c'est foo !

// Les guillemets simples annulent le remplacement des variables
echo 'foo vaut $foo'; // foo vaut $foo

// Si vous n'utilisez pas d'autres caractères,
// vous pouvez afficher plusieurs variables
// en les séparant par des virgules
echo $foo;           // foobar
echo $foo,$bar;     // foobarbarbaz

// Des personnes préfèrent passer plusieurs
// paramètres en utilisant la concaténation
echo 'Cette ', 'chaîne ', 'a été ', 'faite ', 'avec plusieurs paramètres.', chr(10);
echo 'Cette ' . 'chaîne ' . 'a été ' . 'faite ' . 'à l\'aide de la concaténation.' . "\n";

echo <<<END
Cette syntaxe s'intitule le "here document" et
permet d'afficher plusieurs lignes avec de
l'interpolation de variables. Notez que la fin de
la syntaxe doit apparaître sur une nouvelle ligne,
avec uniquement un point-virgule, et pas d'espace
de plus !
END;

// parce que echo() ne se comporte pas comme une fonction, le code suivant n'est pas valide.
($some_var) ? echo('Oui') : echo('Non');

// Cependant, les lignes suivantes sont valides :
($some_var) ? print 'Oui' : print 'Non'; // print est aussi une structure de langage, mais
// il se comporte comme une fonction, donc,
// il peut être utilisé dans ce contexte.

echo $some_var ? 'Oui' : 'Non';
?>

```

echo dispose aussi d'une version courte, où vous pouvez faire suivre la balise PHP ouvrante d'un signe égal (=). Cette syntaxe n'est possible que si la directive de configuration short_open_tag a été activée.

Afficher sans l'aide de echo

```
J'ai <?=$foo?> fous.
```

Pour une brève discussion à propos des différences entre print et echo, consultez cet article des FAQTs Knowledge Base : http://www.faqs.com/knowledge_base/view.phtml/aid/1/fid/40

Note

Comme ceci est une structure du langage, et non pas une fonction, il n'est pas possible de l'appeler avec les variables fonctions.

Voir aussi print, printf et flush.

8.127.19 explode() : Coupe une chaîne en segments

array **explode** (string separator , string string , int limit)

explode retourne un tableau de chaînes. Ce sont les sous-chaînes extraites de string, en utilisant le séparateur separator. Si limit est fourni, le tableau retourné aura un maximum de limit éléments, et le dernier élément du tableau contiendra le reste de la chaîne string.

Si separator est une chaîne vide (""), `explode` retournera FALSE . Si separator contient une valeur qui n'est pas contenue dans string , alors `explode` retournera un tableau, contenant la chaîne string entière.

Si le paramètre limit a une valeur négative, tous les éléments exceptés le dernier limit sont retournés. Cette fonctionnalité a été ajoutée depuis PHP 5.1.0.

Bien que `implode` puisse, pour des raisons historiques, accepter ces paramètres dans n'importe quel ordre, `explode` ne le peut pas. Vous devez vous assurer que le paramètre separator soit placé avant le paramètre string .

Note

Le paramètre limit a été ajouté en PHP 4.0.1.

Exemple avec `explode`

```
<?php
// Exemple 1
$pizza = "pièce1 pièce2 pièce3 pièce4 pièce5 pièce6";
$pieces = explode(" ", $pizza);
echo $pieces[0]; // pièce1
echo $pieces[1]; // pièce2

// Exemple 2
$data = "foo:*:1023:1000::/home/foo:/bin/sh";
list($user, $pass, $uid, $gid, $gecos, $home, $shell) = explode(":", $data);
echo $user; // foo
echo $pass; // *

?>
```

Exemple avec le paramètre limit

```
<?php
$str = 'un|deux|trois|quatre';

// limite positive
print_r(explode('|', $str, 2));

// limite négative (depuis PHP 5.1)
print_r(explode('|', $str, -1));

?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [0] => un
    [1] => deux|trois|quatre
)
Array
(
    [0] => un
    [1] => deux
    [2] => trois
)
```

Note

Cette fonction gère les chaînes binaires.

Voir aussi [preg_split](#) , [spliti](#) , [split](#) , [strtok](#) , et [implode](#) .

8.127.20 fprintf() : Écrit une chaîne formatée dans un flux

int **fprintf** (resource handle , string format , *mixed args* , *mixed ...*)

[fprintf](#) écrit la chaîne produite avec le format format dans le flux représenté par handle . Le format format est décrit dans la documentation de la fonction [sprintf](#) .

Retourne la longueur de la chaîne affichée.

8.127.21 get_html_translation_table() : Retourne la table de traduction des entités utilisée par [htmlspecialchars](#) et [htmlentities](#)

array **get_html_translation_table** (int table , int quote_style)

[get_html_translation_table](#) retourne la table de traduction des entités utilisée en interne par les fonctions [htmlspecialchars](#) et [htmlentities](#) .

Il existe deux constantes (HTML_ENTITIES et HTML_SPECIALCHARS) , qui vous permettent de spécifier la table que vous souhaitez. Le paramètre table vaut par défaut HTML_SPECIALCHARS . Et comme dans les fonctions [htmlspecialchars](#) et [htmlentities](#) , vous pouvez optionnellement spécifier le type de guillemets que vous voulez utiliser. Le comportement par défaut est le mode ENT_COMPAT . Pour la description de ces modes, reportez-vous à [htmlspecialchars](#) .

Note

Les caractères spéciaux peuvent être encodés de différentes façon. E.g. " peut être encodé comme " , " ou " . get_html_translation_table retourne uniquement la forme la plus courante d'encodage.

Exemple avec la table de traduction des caractères en entités HTML
--

<pre><?php \$trans = get_html_translation_table(HTML_ENTITIES); \$str = "Hallo & <Frau> & Krämer"; \$encoded = strtr(\$str, \$trans); ?></pre>
--

La variable \$encoded contient maintenant "Hallo & <Frau> & & Krämer".

Voir aussi [htmlspecialchars](#) , [htmlentities](#) et [html_entity_decode](#) .

8.127.22 hebrew() : Convertit un texte logique hébreu en texte visuel

string **hebrew** (string hebrew_text , int max_chars_per_line)

hebrew convertit le texte logique hébreu hebrew_text en texte visuel. Le paramètre optionnel max_chars_per_line indique le nombre maximum de caractères par ligne dans le résultat. [hebrew](#) essaie d'éviter de scinder les mots.

Voir aussi [hebrevc](#) .

8.127.23 `hebrevc()` : Convertit un texte logique hébreux en texte visuel, avec retours à la ligne

string `hebrevc` (string `hebrew_text` , int `max_chars_per_line`)

`hebrevc` est similaire à `hebrew` à la différence qu'elle convertit les nouvelles lignes de la chaîne de caractères `hebrew_text` (\n) en " `
\n` ". Le paramètre optionnel `max_chars_per_line` indique le nombre maximum de caractères par ligne dans le résultat. `hebrevc` essaie d'éviter de scinder les mots.

Voir aussi [hebrew](#) .

8.127.24 `html_entity_decode()` : Convertit toutes les entités HTML en caractères normaux

string `html_entity_decode` (string `string` , int `quote_style` , string `charset`)

`html_entity_decode` est la fonction contraire de `htmlentities` : elle convertit les entités HTML de la chaîne string en caractères normaux.

Le paramètre optionnel `quote_style` vous permet de définir ce qu'il adviendra des guillemets simples et doubles. Ce paramètre prend l'une des valeurs suivantes (et la valeur par défaut est `ENT_COMPAT`) :

Constante	Description
<code>ENT_COMPAT</code>	Convertit les guillemets doubles et ignore les guillemets simples.
<code>ENT_QUOTES</code>	Convertit les guillemets doubles et les guillemets simples.
<code>ENT_NOQUOTES</code>	Ne convertit aucun guillemet.

Le jeu de caractères ISO-8859-1 est utilisé par défaut, comme paramètre `charset` . Ce paramètre permet de choisir le jeu de caractères utilisé dans la conversion.

Les jeux de caractères suivants sont disponibles et supportés par PHP 4.3.0 et plus récent.

Jeux de caractères	Alias	Description
ISO-8859-1	ISO8859-1	Europe occidentale, Latin-1
ISO-8859-15	ISO8859-15	Europe occidentale, Latin-9. Dispose du signe Euro, des caractères spéciaux français et finlandais, qui manquent au Latin-1(ISO-8859-1).
UTF-8		Unicode 8 bits multi octets, compatible avec l'ASCII
cp866	ibm866, 866	Jeu de caractères Cyrillic spécifique à DOS. Ce jeu de caractères est supporté depuis PHP 4.3.2.
cp1251	Windows-1251, win-1251, 1251	Jeu de caractères Cyrillic spécifique à Windows. Ce jeu de caractères est supporté depuis PHP 4.3.2.
cp1252	Windows-1252, 1252	Jeu de caractères spécifique de Windows pour l'Europe occidentale.
KOI8-R	koi8-ru, koi8r	Russe. Ce jeu de caractères est supporté depuis PHP 4.3.2.

BIG5	950	Chinois traditionnel, principalement utilisé à Taiwan.
GB2312	936	Chinois simplifié, officiel.
BIG5-HKSCS		Big5 avec les extensions de Hong Kong, chinois traditionnel.
Shift_JIS	SJIS, 932	Japonais
EUC-JP	EUCJP	Japonais
Note		
Les autres jeux de caractères ne sont pas reconnus, et le ISO-8859-1 sera utilisé à la place.		

Note
Cette fonction ne supporte pas les jeux de caractères multi-octets dans PHP < 5.
Décoder des entités HTML
<pre> <?php \$orig = 'J\'ai "sorti" le chien tout à l\'heure'; \$a = htmlentities(\$orig); \$b = html_entity_decode(\$a); echo \$a; // J'ai "sorti" le chien tout à l'heure echo \$b; // J'ai "sorti" le chien tout à l'heure // Pour les utilisateurs ayant des versions antérieures à PHP 4.3.0 : function unhtmlentities (\$string) { // Remplace les entités numériques \$string = preg_replace('~&#x([0-9a-f]+);~ei', 'chr(hexdec("\1"))', \$string); \$string = preg_replace('~&#([0-9]+);~e', 'chr(\1)', \$string); // Remplace les entités littérales \$trans_tbl = get_html_translation_table (HTML_ENTITIES); \$trans_tbl = array_flip (\$trans_tbl); return strtr (\$string, \$trans_tbl); } \$c = unhtmlentities(\$a); echo \$c; // J'ai "sorti" le chien tout à l'heure ?> </pre>

Note
<p>Vous pourriez vous demander pourquoi <code>trim(html_entity_decode('&nbsp;'))</code>; ne réduit pas la chaîne à la chaîne vide. C'est parce que l'entité <code>&nbsp;</code> n'est pas un code ASCII 32 (qui serait supprimé par <code>trim</code>) mais un code ASCII 160 (0xa0) dans le jeu de caractères par défaut ISO 8859-1 .</p> <p>Voir aussi htmlentities , get_html_translation_table , htmlspecialchars et urldecode .</p>

8.127.25 htmlentities() : Convertit tous les caractères éligibles en entités HTML

string **htmlentities** (string string , int quote_style , string charset)

htmlentities est identique à la fonction [htmlspecialchars](#) , sauf que tous les caractères qui ont des équivalents en entités HTML sont effectivement traduits.

Comme [htmlspecialchars](#) , cette fonction prend l'argument optionnel `quote_style` , qui indique comment doivent être traités les guillemets doubles et simples. Vous pouvez utiliser l'une des constantes suivantes la valeur par défaut étant `ENT_COMPAT` :

Nom	Description
ENT_COMPAT	Convertit les guillemets doubles, et ignore les guillemets simples
ENT_QUOTES	Convertit les guillemets doubles et les guillemets simples
ENT_NOQUOTES	Ignore tous les guillemets

Le support du paramètre optionnel `quote_style` a été ajouté en PHP 4.0.3.

Comme `htmlspecialchars`, cette fonction prend un troisième argument optionnel qui définit le jeu de caractères utilisé durant la conversion. Le support de cet argument a été ajouté en PHP 4.1.0. Actuellement, le jeu de caractères ISO-8859-1 est utilisé par défaut.

Les jeux de caractères suivants sont disponibles et supportés par PHP 4.3.0 et plus récent.

Jeux de caractères	Alias	Description
ISO-8859-1	ISO8859-1	Europe occidentale, Latin-1
ISO-8859-15	ISO8859-15	Europe occidentale, Latin-9. Dispose du signe Euro, des caractères spéciaux français et finlandais, qui manquent au Latin-1(ISO-8859-1).
UTF-8		Unicode 8 bits multi octets, compatible avec l'ASCII
cp866	ibm866, 866	Jeu de caractères Cyrillic spécifique à DOS. Ce jeu de caractères est supporté depuis PHP 4.3.2.
cp1251	Windows-1251, win-1251, 1251	Jeu de caractères Cyrillic spécifique à Windows. Ce jeu de caractères est supporté depuis PHP 4.3.2.
cp1252	Windows-1252, 1252	Jeu de caractères spécifique de Windows pour l'Europe occidentale.
KOI8-R	koi8-ru, koi8r	Russe. Ce jeu de caractères est supporté depuis PHP 4.3.2.
BIG5	950	Chinois traditionnel, principalement utilisé à Taiwan.
GB2312	936	Chinois simplifié, officiel.
BIG5-HKSCS		Big5 avec les extensions de Hong Kong, chinois traditionnel.
Shift_JIS	SJIS, 932	Japonais
EUC-JP	EUCJP	Japonais

Note

Les autres jeux de caractères ne sont pas reconnus, et le ISO-8859-1 sera utilisé à la place.

Si vous voulez réaliser l'opération inverse, vous devez utiliser la fonction `html_entity_decode`.

Exemple avec `htmlspecialchars`

```
<?php
$str = 'Un \'apostrophe\' en <strong>gras</strong>';

// Affiche : Un 'apostrophe' en <strong>gras</strong>
echo htmlspecialchars($str);

// Affiche : Un &#039;apostrophe&#039; en <strong>gras</strong>
echo htmlspecialchars($str, ENT_QUOTES);
?>
```

Voir aussi `html_entity_decode`, `get_html_translation_table`, `htmlspecialchars`, `nl2br` et `urlencode`.

8.127.26 htmlspecialchars_decode() : Convertit les entités HTML spéciales en caractères

string **htmlspecialchars_decode** (string string , int quote_style)

Cette fonction est l'opposée de [htmlspecialchars](#) . Elle convertit les entités HTML spéciales en caractères.

Les entités converties sont : & ; , " (lorsque ENT_NOQUOTES n'est pas activée), ' (lorsque ENT_QUOTES est activée), < et > .

8.127.27 htmlspecialchars() : Convertit les caractères spéciaux en entités HTML

string **htmlspecialchars** (string string , int quote_style , string charset)

Certains caractères ont des significations spéciales en HTML, et doivent être remplacés par des entités HTML pour être affichés. [htmlspecialchars](#) remplace tous ces caractères par leur équivalent dans la chaîne string . Cette conversion est très pratique pour la programmation web. Si vous devez remplacer tous les caractères, utilisez plutôt [htmlentities](#) à la place.

[htmlspecialchars](#) est pratique pour éviter que des données fournies par les utilisateurs contiennent des balises HTML, comme pour un forum ou un chat. [htmlspecialchars](#) prend un deuxième argument optionnel quote_style , qui indique comment doivent être traités les guillemets doubles et simples. Vous pouvez utiliser l'une des constantes suivantes : ENT_COMPAT , la constante par défaut, va convertir les guillemets doubles et ignorer les guillemets simples; ENT_QUOTES va convertir les guillemets doubles et les guillemets simples; ENT_NOQUOTES va ignorer les guillemets doubles et les guillemets simples.

Les remplacements effectués sont :

- " & " (et commercial) devient " & ; "
- " " " (guillemets doubles) devient " " ; " lorsque ENT_NOQUOTES n'est pas utilisée.
- " ' " (single quote) devient " ' ; " uniquement lorsque ENT_QUOTES est utilisée.
- " < " (supérieur à) devient " < ; "
- " > " (supérieur à) devient " > ; "

Exemple avec htmlspecialchars

```
<?php
$new = htmlspecialchars("<a href='test'>Test</a>", ENT_QUOTES);
echo $new;
// <a href=&#039;test&#039;>Test</a>
?>
```

Notez que cette fonction ne fait aucun autre remplacement que ceux qui sont listés ci-dessus. Pour faire un remplacement total, voyez plutôt [htmlentities](#) . Le support pour le second argument optionnel a été ajouté en PHP 3.0.17 et PHP 4.0.3.

Cette fonction prend un troisième argument optionnel qui définit le jeu de caractères utilisé durant la conversion. Le jeu de caractères par défaut est ISO-8859-1. Le support de cet argument a été ajouté en PHP 4.1.0.

Les jeux de caractères suivants sont disponibles et supportés par PHP 4.3.0 et plus récent.

Jeux de caractères	Alias	Description
ISO-8859-1	ISO8859-1	Europe occidentale, Latin-1
ISO-8859-15	ISO8859-15	Europe occidentale, Latin-9. Dispose du signe Euro, des caractères spéciaux français et finlandais, qui manquent au Latin-1 (ISO-8859-1).
UTF-8		Unicode 8 bits multi octets, compatible avec l'ASCII
cp866	ibm866, 866	Jeu de caractères Cyrillic spécifique à DOS. Ce jeu de caractères est supporté depuis PHP 4.3.2.
cp1251	Windows-1251, win-1251, 1251	Jeu de caractères Cyrillic spécifique à Windows. Ce jeu de caractères est supporté depuis PHP 4.3.2.
cp1252	Windows-1252, 1252	Jeu de caractères spécifique de Windows pour l'Europe occidentale.
KOI8-R	koi8-ru, koi8r	Russe. Ce jeu de caractères est supporté depuis PHP 4.3.2.
BIG5	950	Chinois traditionnel, principalement utilisé à Taiwan.
GB2312	936	Chinois simplifié, officiel.
BIG5-HKSCS		Big5 avec les extensions de Hong Kong, chinois traditionnel.
Shift_JIS	SJIS, 932	Japonais
EUC-JP	EUCJP	Japonais
Note		
Les autres jeux de caractères ne sont pas reconnus, et le ISO-8859-1 sera utilisé à la place.		

Voir aussi [get_html_translation_table](#) , [strip_tags](#) , [htmlentities](#) et [nl2br](#) .

8.127.28 implode() : Rassemble les éléments d'un tableau en une chaîne

string **implode** (string glue , array pieces)

implode retourne une chaîne contenant la représentation en chaîne de caractères de tous les éléments du tableau `pieces` , dans le même ordre, avec la chaîne `glue` , placée entre deux éléments.

Exemple avec <code>implode</code>
<pre><?php \$array = array('nom', 'email', 'telephone'); \$comma_separated = implode(", ", \$array); echo \$comma_separated; // nom,email,telephone ?></pre>
Note
<code>implode</code> peut, pour des raisons historiques, accepter les paramètres dans un sens ou dans l'autre. Pour des raisons de cohérence avec <code>explode</code> , toutefois, il est préférable d'utiliser l'ordre des arguments tels que documenté.
Note
Depuis PHP 4.3.0, le paramètre <code>glue</code> de <code>implode</code> est optionnel et, par défaut, il vaut la chaîne vide " . Ce n'est pas l'utilisation recommandée de la fonction <code>implode</code> . Nous vous recommandons de toujours utiliser les deux paramètres pour être compatible avec les anciennes versions.
Note
Cette fonction gère les chaînes binaires.

Voir aussi [explode](#) et [split](#) .

8.127.29 join() : Alias de implode

Cette fonction est un alias de : implode .

8.127.30 levenshtein() : Calcule la distance Levenshtein entre deux chaînes

int **levenshtein** (string str1 , string str2 , int cost_ins , int cost_rep , int cost_del)

levenshtein calcule la distance Levenshtein entre deux chaînes de caractères. Elle retournera -1 si l'un des deux arguments contient plus de 255 caractères.

La distance Levenshtein est définie comme le nombre minimal de caractères qu'il faut remplacer, insérer ou modifier pour transformer la chaîne str1 en str2 . La complexité de l'algorithme est en $O(m*n)$, où n et m sont les tailles respectives de str1 et str2 : c'est plutôt bien, en comparaison de similar_text , qui est en $O(\max(n,m)**3)$, mais cela reste très coûteux.

Dans sa forme la plus simple, levenshtein va prendre uniquement deux chaînes de caractères comme paramètres, et calculer simplement le nombre d'insertions, de remplacements et d'effacements nécessaires pour transformer str1 en str2 .

La deuxième variante de la fonction prend trois paramètres supplémentaires qui représentent les coûts d'insertions, de remplacements et d'effacements. C'est une version plus générale de la première fonction, mais qui est un peu moins efficace.

Exemple avec levenshtein

```

<?php
// mot mal orthographié
$input = 'carrrot';

// tableau de mots à vérifier
$words = array('apple','pineapple','banana','orange',
               'radish','carrot','pea','bean','potato');

// aucune distance de trouvée pour le moment
$shortest = -1;

// boucle sur les des mots pour trouver le plus près
foreach ($words as $word) {

    // calcule la distance avec le mot mis en entrée,
    // et le mot courant
    $lev = levenshtein($input, $word);

    // cherche une correspondance exacte
    if ($lev == 0) {

        // le mot le plus près est celui-ci (correspondance exacte)
        $closest = $word;
        $shortest = 0;

        // on sort de la boucle ; nous avons trouvé une correspondance exacte
        break;
    }

    // Si la distance est plus petite que la prochaine distance trouvée
    // OU, si le prochain mot le plus près n'a pas encore été trouvé
    if ($lev <= $shortest || $shortest < 0) {

```

```

        // définition du mot le plus près ainsi que la distance
        $closest = $word;
        $shortest = $lev;
    }
}

echo "Mot entré : $input\n";
if ($shortest == 0) {
    echo "Correspondance exacte trouvée : $closest\n";
} else {
    echo "Vous voulez dire : $closest ?\n";
}
?>

```

L'exemple ci-dessus va afficher :

```

Mot entré : carrot
Vous voulez dire : carrot ?

```

Voir aussi [soundex](#) , [similar_text](#) et [metaphone](#) .

8.127.31 localeconv() : Lit la configuration locale

array **localeconv** (void)

localeconv retourne un tableau associatif contenant les informations de formats localisées pour les nombres et la monnaie.

localeconv retourne les formats en fonction de la configuration effectuée avec [setlocale](#) . Le tableau associatif qui est retourné contient les index suivants :

Index du tableau	Description
decimal_point	Séparateur décimal
thousands_sep	Séparateur de milliers
grouping	Tableau contenant les regroupements numériques
int_curr_symbol	Symbole monétaire international (i.e. EUR)
currency_symbol	Symbole monétaire local (i.e. €)
mon_decimal_point	Séparateur décimal monétaire
mon_thousands_sep	Séparateur de milliers monétaires
mon_grouping	Tableau contenant les regroupements numériques monétaires
positive_sign	Signe des valeurs positives
negative_sign	Signe des valeurs négatives
int_frac_digits	Nombre international de décimales
frac_digits	Nombre local de décimales
p_cs_precedes	TRUE si le currency_symbol précède une valeur positive et FALSE s'il lui succède.
p_sep_by_space	TRUE si un espace sépare currency_symbol d'une valeur positive, et FALSE sinon.
n_cs_precedes	TRUE si currency_symbol précède une valeur négative, et FALSE s'il lui succède.

n_sep_by_space	TRUE si un espace sépare currency_symbol d'une valeur négative, et FALSE sinon.
p_sign_posn	<ul style="list-style-type: none"> • 0 - Des parenthèses entourent la valeur et le symbole monétaire • 1 - Le signe précède la valeur et le symbole monétaire • 2 - Le signe suit la valeur et le symbole monétaire • 3 - Le signe précède immédiatement la valeur et le symbole monétaire • 4 - Le signe suit immédiatement la valeur et le symbole monétaire
n_sign_posn	<ul style="list-style-type: none"> • 0 - Des parenthèses entourent la valeur et le symbole monétaire • 1 - Le signe précède la valeur et le symbole monétaire • 2 - Le signe suit la valeur et le symbole monétaire • 3 - Le signe précède immédiatement la valeur et le symbole monétaire • 4 - Le signe suit immédiatement la valeur et le symbole monétaire

Les champs p_sign_posn et n_sign_posn contiennent une chaîne formatée d'options. Chaque nombre représente une des conditions listées ci-dessus.

Les champs de regroupements contiennent des tableaux qui définissent la manière dont les nombres doivent être regroupés. Par exemple, le champ de regroupement monétaire pour nl_NL (en mode UTF-8 avec le signe euro), contiendra deux éléments, avec les valeurs 3 et 3 . Si un élément de tableau contient CHAR_MAX , aucun autre regroupement n'est fait. Si un élément de tableau contient 3 , l'élément précédent doit être utilisé.

Exemple avec localeconv

```
<?php
if (false !== setlocale(LC_ALL, 'fr_FR@euro')) {
    $locale_info = localeconv();
    print_r($locale_info);
}
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [decimal_point] => ,
    [thousands_sep] =>
    [int_curr_symbol] => EUR
    [currency_symbol] => ¤
    [mon_decimal_point] => ,
    [mon_thousands_sep] =>
    [positive_sign] =>
    [negative_sign] => -
    [int_frac_digits] => 2
    [frac_digits] => 2
    [p_cs_precedes] => 0
    [p_sep_by_space] => 1
    [n_cs_precedes] => 0
    [n_sep_by_space] => 1
    [p_sign_posn] => 1
    [n_sign_posn] => 1
```

```
[grouping] => Array
(
)

[mon_grouping] => Array
(
  [0] => 3
  [1] => 3
)
)
```

Voir aussi [setlocale](#) .

8.127.32 ltrim() : Supprime les espaces (ou d'autres caractères) de début de chaîne

string **ltrim** (string *str* , string *charlist*)

[ltrim](#) retourne la chaîne *str* , après avoir supprimé les caractères invisibles de début de chaîne. Si le second paramètre *charlist* a été omis, [ltrim](#) supprimera les caractères suivants :

- " " (ASCII 32 (0x20)), un espace ordinaire.
- "\t" (ASCII 9 (0x09)), une tabulation.
- "\n" (ASCII 10 (0x0A)), une nouvelle ligne (line feed).
- "\r" (ASCII 13 (0x0D)), un retour chariot (carriage return).
- "\0" (ASCII 0 (0x00)), le caractère NUL .
- "\x0B" (ASCII 11 (0x0B)), une tabulation verticale.

Il est aussi possible de spécifier les caractères à supprimer en utilisant le paramètre *charlist* . Listez simplement les caractères que vous voulez supprimer dans ce paramètre. Avec .. , vous pourrez spécifier des intervalles de caractères.

Exemple avec [ltrim](#)

```
<?php
$text = "\t\tVoici quelques mots :) ... ";
$trimmed = ltrim($text);
// $trimmed = "Voici quelques mots :) ... "
$trimmed = ltrim($text, " \t\r\n");
// $trimmed = "Voici quelques mots :) ... "
$clean = ltrim($binary, "\0x00..\x1F");
// supprime tous les caractères de contrôle ASCII au début de la chaîne de caractères.
// (de 0 à 31 inclus)
?>
```

Note

Le second paramètre a été ajouté en PHP 4.1.0.

Voir aussi [trim](#) et [rtrim](#) .

8.127.33 md5_file() : Calcule le md5 d'un fichier

string **md5_file** (string *filename* , bool *raw_output*)

`md5_file` calcule le MD5 du fichier filename en utilisant l'algorithme [RSA Data Security, Inc. MD5 Message-Digest Algorithm](#) , puis retourne la valeur ainsi calculée. Le résultat est un nombre de 32 caractères hexadécimaux.

8.127.34 md5() : Calcule le md5 d'une chaîne

string `md5` (string `str` , bool `raw_output`)

`md5` calcule le MD5 de la chaîne de caractères `str` en utilisant l'algorithme [RSA Data Security, Inc. MD5 Message-Digest Algorithm](#) , et retourne le résultat. Le résultat est un nombre de 32 caractères hexadécimaux.

Note

Le paramètre `raw_output` a été ajouté en PHP 5.0.0 et vaut par défaut FALSE .

Exemple avec `md5`

```
<?php
$str = 'pomme';

if (md5($str) === 'ede0f9c3a1d2093e3f48fcafd3c70915') {
    echo "Voulez-vous une golden ou une spartan?";
    exit;
}
?>
```

Voir aussi [crc32](#) , [md5_file](#) et [sha1](#) .

8.127.35 metaphone() : Calcule la clé metaphone

string `metaphone` (string `str` , int `phones`)

`metaphone` calcule la clé metaphone de la chaîne de caractères `str` .

`metaphone` est similaire à la fonction `soundex` : elle crée une clé similaire pour des mots dont la prononciation est proche. C'est une fonction qui est plus précise que `soundex` car elle prend en compte la prononciation anglaise. La clé metaphone générée est de taille variable.

`metaphone` fonctionne en anglais, et n'utilise pas les spécificités de la langue française. Les résultats sur des termes français peuvent être des plus aléatoires.

Metaphone a été développée par Lawrence Philips philips@verity.com . Cette méthode est décrite dans le livre ["Practical Algorithms for Programmers", Binstock & Rex, Addison Wesley, 1995] .

8.127.36 money_format() : Met un nombre au format monétaire

string `money_format` (string `format` , float `number`)

`money_format` retourne le nombre `number` avec le format financier. Cette fonction fait l'interface avec la fonction `strfmon()` de la bibliothèque C, à la différence près que cette implémentation ne convertit qu'un nombre à la fois.

Note

La fonction `money_format` est uniquement définie si le système a les capacités `strfmon`. Par exemple, Windows ne les a pas, donc, `money_format` n'est pas définie sous Windows.

Le paramètre de format est constitué de la séquence suivante :

- un caractère %
- une configuration optionnelle
- une taille de champ optionnelle
- une précision à gauche optionnelle
- une précision à droite optionnelle
- un caractère de conversion obligatoire

Configuration

Une ou plusieurs des configurations suivantes sont utilisables :

= f

Le caractère = suivi par un octet unique f qui sera utilisé comme caractère de remplissage. Le caractère de remplissage par défaut est espace.

^

Désactive le groupage de caractères (tel que défini dans la configuration locale).

+ ou (

Spécifie le style de formatage pour les nombres positifs et négatifs. Si + est utilisé, les équivalents dans la configuration locale de + et - seront utilisés. Si (est utilisé, les sommes négatives seront placées entre parenthèses. Si aucune spécification n'est fournie, la valeur par défaut est + .

!

Supprime le simple monétaire dans la chaîne finale.

-

Si fourni, cette configuration fait que les champs seront justifiés à gauche (complétés à droite), au contraire de la configuration par défaut qui est justifiée à droite, et complétée à gauche.

Taille de champ

w

Un nombre décimal qui spécifie la taille minimale du champ. Le champ sera complété à gauche, à moins que la configuration - ne soit utilisée. Par défaut, cette valeur est de 0.

Précision à gauche

n

Le nombre maximal de chiffres (n) attendus à gauche du séparateur décimal (e.g. la virgule). Cette option est généralement utilisée pour conserver l'alignement de colonnes de nombres, en utilisant un caractère pour compléter le nombre si ce dernier a moins de n chiffres. Si le nombre réel de chiffres est plus grand que n , cette spécification est ignorée.

Si le groupage n'a pas été supprimé via la configuration ^ , les séparateurs de groupage seront insérés avant le caractère de remplissage (le cas échéant). Les séparateurs ne seront pas appliqués aux caractères de remplissage, même si ce caractère est un nombre.

Pour s'assurer de l'alignement, tous les caractères apparaissant avant et après le nombre formaté, tels que les symboles monétaires ou les signes négatif et positif, seront placés au même endroit grâce à des espaces supplémentaires, afin que toutes les tailles des nombres soient les mêmes.

Précision à droite

.p

Un point suivi par un nombre de décimales (p). Si la valeur de p est 0 (zéro), le séparateur décimal et les décimales seront supprimés. Si aucune précision à droite n'est précisée, la valeur par défaut sera lue dans la configuration locale. Le nombre formaté sera alors arrondi pour satisfaire les contraintes d'affichage.

Caractères de conversion

i

Le nombre est formaté suivant le format monétaire international de la configuration locale (e.g. pour la France : 1 234,56 F).

n

Le nombre est formaté en fonction du format monétaire national (e.g. pour la configuration de_DE : DM1.234,56).

%

Retourne le caractère % .

Note

La catégorie LC_MONETARY de la configuration locale affecte le comportement de cette fonction. Utilisez [setlocale](#) pour configurer correctement PHP avant d'utiliser cette fonction.

Les caractères placés avant et après la chaîne de formatage seront laissés intacts.

Exemple avec `money_format`

Voici plusieurs exemples d'utilisation de la fonction `money_format` avec différentes chaînes de formatage, et configurations locales.

```
<?php
$number = 1234.56;

// Affichons ce nombre au format international pour en_US
setlocale(LC_MONETARY, 'en_US');
echo money_format('%i', $number)."\n";
// USD 1,234.56

// Et au format italien national avec 2 décimales
setlocale(LC_MONETARY, 'it_IT');
echo money_format('%.2n', $number)."\n";
// L. 1.234,56

// Utilisation d'un nombre négatif
$number = -1234.5672;

// Format US national, avec les parenthèses pour les nombres négatifs
// et 10 chiffres de précision à gauche
setlocale(LC_MONETARY, 'en_US');
echo money_format('%(#10n', $number)."\n";
// ($          1,234.57)

// Format similaire au précédent, en ajoutant 2 décimales
```

```
// pour la précision à droite, et en utilisant le caractère de remplissage '*'
echo money_format('%=#10.2n', $number)."\n";
// ($*****1,234.57)

// Utilisons maintenant la justification à gauche, avec un champ de 14 caractères
// de long, sans groupage de chiffres, et en utilisant le format international
// pour de_DE
setlocale(LC_MONETARY, 'de_DE');
echo money_format('%=*^-14#8.2i', 1234.56)."\n";
// DEM 1234,56****

// Ajoutons encore à l'exemple précédent
setlocale(LC_MONETARY, 'en_GB');
$fmt = 'La valeur finale est %i (après 10 % de remise)';
echo money_format($fmt, 1234.56)."\n";
// La valeur finale est GBP 1,234.56 (après 10 % de remise)

?>
```

Voir aussi [setlocale](#) , [number_format](#) , [sprintf](#) , [printf](#) et [scanf](#) .

8.127.37 nl_langinfo() : Rassemble des informations sur la langue et la configuration locale

string **nl_langinfo** (int item)

[nl_langinfo](#) est utilisée pour accéder à chaque élément de la configuration locale. Contrairement à la fonction [localeconv](#) qui renvoie tous les éléments, [nl_langinfo](#) vous permet de sélectionner un élément précis.

Si le paramètre item n'est pas valide, FALSE sera retourné.

item peut être la valeur entière d'un élément, ou le nom de sa constante. Voici une liste des noms de constantes pour item qui peuvent être utilisés et leur description. Certaines constantes peuvent ne pas être définies, ou ne contenir aucune valeur pour certaines locales.

Constante	Description
Constantes de la catégorie LC_TIME	
ABDAY_(1-7)	Nom court du n-ème jour de la semaine.
DAY_(1-7)	Nom du n-ème jour de la semaine (DAY_1 = Dimanche).
ABMON_(1-12)	Nom abrégé du n-ème mois de l'année.
MON_(1-12)	Nom du n-ème mois de l'année.
AM_STR	Chaîne pour Ante meridian.
PM_STR	Chaîne pour Post meridian.
D_T_FMT	Chaîne pouvant être utilisée comme chaîne de formatage pour la fonction strftime pour représenter la date et l'heure.
D_FMT	Chaîne pouvant être utilisée comme chaîne de formatage pour la fonction strftime pour représenter la date.
T_FMT	Chaîne pouvant être utilisée comme chaîne de formatage pour la fonction strftime pour représenter l'heure.
T_FMT_AMPM	Chaîne pouvant être utilisée comme chaîne de formatage pour la fonction strftime pour représenter l'heure au format 12-heure avec ante/post meridian.

ERA	Alternative era.
ERA_YEAR	Année dans le format alternatif era.
ERA_D_T_FMT	Date et heure dans le format alternatif era (chaîne pouvant être utilisée dans la fonction <u>strftime</u>).
ERA_D_FMT	Date dans le format alternatif era (chaîne pouvant être utilisée dans la fonction <u>strftime</u>).
ERA_T_FMT	Heure dans le format alternatif era (chaîne pouvant être utilisée dans la fonction <u>strftime</u>).
Constantes de la catégorie LC_MONETARY	
INT_CURR_SYMBOL	Symbole monétaire international.
CURRENCY_SYMBOL	Symbole monétaire local.
CRNCYSTR	Même valeur que CURRENCY_SYMBOL.
MON_DECIMAL_POINT	Caractère de virgule décimale.
MON_THOUSANDS_SEP	Séparateur de centaine (groupes de trois lettres).
MON_GROUPING	Comme l'élément 'grouping'.
POSITIVE_SIGN	Signe pour les valeurs positives.
NEGATIVE_SIGN	Signe pour les valeurs négatives.
INT_FRAC_DIGITS	Chiffres partiels internationaux.
FRAC_DIGITS	Chiffres partiels locaux.
P_CS_PRECEDES	Retourne 1 si CURRENCY_SYMBOL précède une valeur positive.
P_SEP_BY_SPACE	Retourne 1 si un espace sépare CURRENCY_SYMBOL d'une valeur positive.
N_CS_PRECEDES	Retourne 1 si CURRENCY_SYMBOL précède une valeur négative.
N_SEP_BY_SPACE	Retourne 1 si un espace sépare CURRENCY_SYMBOL d'une valeur négative.
P_SIGN_POSN	<ul style="list-style-type: none"> • Retourne 0 si des parenthèses entourent la quantité et currency_symbol. • Retourne 1 si la chaîne de signes précède la quantité et currency_symbol. • Retourne 2 si la chaîne de signes suit la quantité et currency_symbol. • Retourne 3 si la chaîne de signes précède immédiatement le currency_symbol. • Retourne 4 si la chaîne de signes suit immédiatement le currency_symbol.
N_SIGN_POSN	
Constantes de la catégorie LC_NUMERIC	
DECIMAL_POINT	Caractère de virgule décimale.
RADIXCHAR	Même valeur que DECIMAL_POINT.
THOUSANDS_SEP	Caractère de séparation des centaines (groupe de trois lettres).
THOUSEP	Même valeur que THOUSANDS_SEP.
GROUPING	

Constantes de la catégorie LC_MESSAGES	
YESEXPR	Chaîne regex pour chercher l'entrée " yes ".
NOEXPR	Chaîne regex pour chercher l'entrée " no ".
YESSTR	Affichage de la chaîne pour " yes ".
NOSTR	Affichage de la chaîne pour " no ".
Constantes de la catégorie LC_CTYPE	
CODESET	Retourne une chaîne de caractères avec le nom du jeux de caractères.
Note	
Cette fonction n'est pas implémentée sous Windows.	

Voir aussi [setlocale](#) et [localeconv](#) .

8.127.38 nl2br() : Insère un retour à la ligne HTML à chaque nouvelle ligne

string **nl2br** (string string)

[nl2br](#) retourne string après avoir inséré '
' devant toutes les nouvelles lignes.

Note
Depuis PHP 4.0.5, nl2br est désormais compatible XHTML. Toutes les versions antérieures à 4.0.5 retourneront l'argument string avec ' ' inséré devant les nouvelles lignes, au lieu de ' '.

Exemple avec nl2br
<pre><?php echo nl2br("foo n'est pas\n bar"); ?></pre> <p>L'exemple ci-dessus va afficher :</p> <pre>foo n'est pas
 bar</pre>

Voir aussi [htmlspecialchars](#) , [htmlentities](#) , [wordwrap](#) et [str_replace](#) .

8.127.39 number_format() : Formate un nombre pour l'affichage

string **number_format** (float number , int decimals , string dec_point , string thousands_sep)

[number_format](#) retourne une chaîne représentant number formaté. [number_format](#) accepte un, deux ou 4 paramètres (mais pas trois).

Si seul le paramètre number est donné, il sera formaté sans partie décimale, mais avec une virgule entre chaque millier.

Si les deux paramètres number et decimals sont fournis, number sera formaté avec decimals décimales, un point (".") comme séparateur décimal et une virgule entre chaque millier.

Avec quatre paramètres, `number` sera formaté avec `decimals` décimales, `dec_point` comme séparateur décimal, et `thousands_sep` comme séparateur de milliers.

Seul le premier caractère du paramètre `thousands_sep` est utilisé. Par exemple, si vous utilisez `bar` comme séparateur de milliers, sur le nombre 1000, `number_format` retournera 1b000.

Exemple avec `number_format`

En notation française, on utilise généralement deux chiffres après la virgule, une virgule comme séparateur décimal, et un espace comme séparateur de milliers. Cela donne :

```
<?php
$nombre = 1234.56;

// Notation anglaise (par défaut)
$english_format_number = number_format($nombre);
// 1,235

// Notation française
$nombre_format_francais = number_format($nombre, 2, ',', ' ');
// 1 234,56

$nombre = 1234.5678;

// Notation anglaise sans séparateur des centaines
$english_format_number = number_format($nombre, 2, '.', '');
// 1234.57

?>
```

Voir aussi `sprintf`, `printf` et `scanf`.

8.127.40 `ord()` : Retourne le code ASCII d'un caractère

int `ord` (string string)

`ord` retourne le code ASCII du premier caractère de `string`. Cette fonction est le contraire de `chr`.

Exemple avec `ord`

```
<?php
if (ord($str) == 10) {
    echo "Le premier caractère de \$str est une nouvelle ligne\n";
}
?>
```

Vous pouvez trouver une table des codes ASCII ici : <http://www.asciitable.com>.

Voir aussi `chr`.

8.127.41 `parse_str()` : Analyse une requête HTTP

void `parse_str` (string str , array arr)

`parse_str` analyse la chaîne de caractères `str` comme s'il s'agissait d'une requête HTTP, passée via l'URL. Toutes les variables qu'elle y repère sont alors créées, avec leurs valeurs respectives. Si le second paramètre `arr` est fourni, les variables y seront stockées, sous forme d'index de tableau.

Note

Le support du paramètre optionnel `arr` a été ajouté en PHP 4.0.3.

Note

Pour accéder à l'URL appelante `QUERY_STRING`, vous devez utiliser la variable `$_SERVER['QUERY_STRING']`. Il est aussi intéressant de lire la section sur les [variables externes à PHP](#).

Note

La configuration de `magic_quotes_gpc` affecte l'affichage de cette fonction car `parse_str` utilise le même mécanisme que PHP utilise pour propager les variables `$_GET`, `$_POST`, etc.

Exemple avec `parse_str`

```
<?php
$str = "first=value&arr[]=foo+bar&arr[]=baz";
parse_str($str);
echo $first; // value
echo $arr[0]; // foo bar
echo $arr[1]; // baz

parse_str($str, $output);
echo $output['first']; // value
echo $output['arr'][0]; // foo bar
echo $output['arr'][1]; // baz
?>
```

Voir aussi [parse_url](#), [pathinfo](#), [get_magic_quotes_gpc](#) et [urldecode](#).

8.127.42 `print()` : Affiche une chaîne de caractères

int `print` (string `arg`)

`print` affiche la chaîne de caractères `arg`. `print` retourne toujours 1.

`print` n'est pas vraiment une fonction (c'est techniquement une structure de langage). Cela fait que vous n'êtes pas obligé d'utiliser des parenthèses.

Exemples avec `print`

```
<?php
print("Bonjour le monde");

print "print() fonctionne aussi sans les parenthèses.";

print "Ce print() se
répartit sur plusieurs lignes. Il affiche aussi les
nouvelles lignes";

print "Ce print() se\nrépartit sur plusieurs lignes. Il affiche aussi les\nnouvelles lignes";

print "L'échappement de caractères se fait : \"comme ceci\".";

// Vous pouvez utiliser des variables avec print
$foo = "foobar";
$bar = "barbaz";

print "foo vaut $foo"; // foo vaut foobar

// Vous pouvez aussi utiliser des tableaux
$bar = array("clé" => "foo");
```

```

print "c'est {$bar['clé']} !"; // c'est foo !

// Les guillemets simples annulent le remplacement des variables
print 'foo vaut $foo'; // foo vaut $foo

// Si vous n'utilisez pas d'autres caractères, vous
// pouvez afficher simplement vos variables comme ceci
print $foo; // foobar

print <<<END
Cette syntaxe s'intitule le "here document" et
permet d'afficher plusieurs lignes avec de
l'interpolation de variables. Notez que la fin de
la syntaxe doit apparaître sur une nouvelle ligne,
avec uniquement un point-virgule, et pas d'espace
de plus !
END;
?>

```

Pour une brève discussion à propos des différences entre `print` et `echo` , consultez cet article des FAQTs Knowledge Base : http://www.faqs.com/knowledge_base/view.phtml/aid/1/fid/40

Note

Comme ceci est une structure du langage, et non pas une fonction, il n'est pas possible de l'appeler avec les variables fonctions .

Voir aussi `echo` , `printf` et `flush` .

8.127.43 printf() : Affiche une chaîne de caractères formatée

int `printf` (string format , *mixed args* , *mixed ...*)

`printf` affiche les arguments args formatés sous forme de chaîne de caractères grâce à args . Les options de formatage sont documentées avec la fonction `sprintf` .

Retourne la longueur de la chaîne retournée.

Voir aussi `print` , `sprintf` , `vprintf` , `sscanf` , `fscanf` et `flush` .

8.127.44 quoted_printable_decode() : Convertit une chaîne quoted-printable en chaîne 8 bits

string `quoted_printable_decode` (string str)

`quoted_printable_decode` retourne la chaîne de caractères str , après l'avoir convertie du format quoted printable binaire 8 bits (en accord avec la [RFC2045](#) , section 6.7, et non pas la [RFC2821](#) , section 4.5.2, pour que les virgules additionnelles ne soient pas effacées du début de la ligne). Cette fonction est similaire à `imap_qprint` , hormis le fait qu'elle n'exige pas le module IMAP pour fonctionner.

8.127.45 quotemeta() : Echappe les méta-caractères

string `quotemeta` (string str)

quotemeta retourne la chaîne *str* après avoir introduit un anti-slash (\) devant tous les caractères suivants : . \ + * ? [^] (\$)

Note

Cette fonction gère les chaînes binaires.

Voir aussi [addslashes](#) , [htmlentities](#) , [htmlspecialchars](#) , [nl2br](#) et [stripslashes](#) .

8.127.46 rtrim() : Supprime les espaces (ou d'autres caractères) de fin de chaîne

string **rtrim** (string *str* , string *charlist*)

rtrim retourne la chaîne *str* , après avoir supprimé tous les caractères blancs de fin de chaîne. Appelée sans le second paramètre, **rtrim** supprimera les caractères suivants :

- " " (ASCII 32 (0x20)), un espace ordinaire.
- "\t" (ASCII 9 (0x09)), une tabulation.
- "\n" (ASCII 10 (0x0A)), une nouvelle ligne (line feed).
- "\r" (ASCII 13 (0x0D)), un retour chariot (carriage return).
- "\0" (ASCII 0 (0x00)), le caractère NUL .
- "\x0B" (ASCII 11 (0x0B)), une tabulation verticale.

Il est aussi possible de spécifier les caractères à supprimer en utilisant le paramètre *charlist* . Listez simplement les caractères que vous voulez supprimer dans ce paramètre. Avec .. , vous pourrez spécifier des intervalles de caractères.

Exemple avec rtrim

<pre><?php \$text = " Voici quelques mots :) ... "; \$trimmed = rtrim(\$text); // \$trimmed = " Voici quelques mots :) ... " \$trimmed = rtrim(\$text, " ."); // \$trimmed = " Voici quelques mots :)" \$clean = rtrim(\$binary, "\0x00..\x1F"); // supprime tous les caractères de contrôle ASCII au début de la chaîne de caractères. // (de 0 à 31 inclus) ?></pre>

Note

Le second paramètre a été ajouté en PHP 4.1.0.
--

Voir aussi [trim](#) et [ltrim](#) .

8.127.47 setlocale() : Modifie les informations de localisation

string **setlocale** (int *category* , string *locale* , string ...) string **setlocale** (int *category* , array *locale*)

setlocale retourne la nouvelle configuration locale, ou FALSE si la localisation n'est pas implémentée sur votre plate-forme, si la variable de localisation n'existe pas, ou si la catégorie spécifiée n'est pas valide. Un nom de catégorie invalide générera un message d'alerte. La liste des noms de locales/catégories peut être trouvée en consultant la [RFC 1766](#) ainsi que l' [ISO 639](#) . Les

différentes plates-formes possèdent des conventions de nommages différentes.

`category` est une constante (ou une chaîne) qui spécifie la catégorie de fonctions affectées par la configuration de localisation :

- `LC_ALL` pour toutes les constantes suivantes
- `LC_COLLATE` pour la comparaison de chaînes de caractères. Voir [strcoll](#)
- `LC_CTYPE` pour la classification et la conversion de caractères. Voir [strtoupper](#)
- `LC_MONETARY` pour [localeconv](#)
- `LC_NUMERIC` pour le séparateur décimal. Voir [localeconv](#)
- `LC_TIME` pour le format de date et d'heure avec [strftime](#)
- `LC_MESSAGES` pour les réponses système (disponible si PHP a été compilé avec `libintl`)

Note

Depuis PHP 4.2.0, passer `category` en tant que chaîne est déconseillé, utilisez les constantes mentionnées plus haut à la place. Les passer en tant que chaînes (entre guillemets) provoquera l'affichage d'un message d'avertissement.

Si locale est NULL ou la chaîne vide "", les noms de locales seront pris dans l'environnement, à partir des variables de même nom que les catégories ci-dessus, ou depuis "LANG".

Si locale est NULL ou vaut "0", la configuration locale ne sera pas modifiée, et la configuration courante sera retournée.

Si locale est un tableau ou bien est suivi par des paramètres additionnels, alors chaque élément du tableau ou chaque paramètre tente d'être défini comme nouvelle locale jusqu'à ce qu'un réussisse. Cela est pratique si la locale est connue sous différents noms sur des systèmes différents ou bien pour prévoir une autre valeur en cas de non disponibilité de la locale choisie.

Note

La possibilité de passer plusieurs locales est disponible depuis PHP 4.3.0.

Note

La valeur retournée par [setlocale](#) dépend du système sur lequel PHP est installé. [setlocale](#) retourne exactement ce que la fonction système `setlocale()` retourne.

Attention

L'information locale est maintenue par processus, non par thread. Si vous faites fonctionner PHP sur un serveur multi-threadé comme IIS ou Apache sur Windows, vous pourriez obtenir des changements soudains des configurations locales pendant qu'un script fonctionne, même si celui-ci n'appelle jamais la fonction [setlocale](#) . Ceci survient à cause des autres scripts qui fonctionnent dans des threads différents du même processus. Ces scripts changent les configurations locales dans le processus au complet en utilisant la fonction [setlocale](#) .

Array

Exemple avec [setlocale](#)

```
<?php
/* Configure le script en hollandais */
setlocale (LC_ALL, 'nl_NL');

/* Affiche : vrijdag 22 december 1978 */
echo strftime("%A %e %B %Y", mktime(0, 0, 0, 12, 22, 1978));

/* Essai de différentes valeurs possible pour l'allemand depuis PHP 4.3.0 */
$loc_de = setlocale(LC_ALL, 'de_DE@euro', 'de_DE', 'de', 'ge');
echo "L'identifiant de l'allemand sur ce système est '$loc_de'";
?>
```

Exemple avec setlocale pour Windows

```
<?php
/* Configure le script en hollandais */
setlocale(LC_ALL, 'nld_nld');

/* Affiche : vrijdag 22 december 1978 */
echo strftime("%A %d %B %Y", mktime(0, 0, 0, 12, 22, 1978));

/* Essai de différentes valeurs possible pour l'allemand depuis PHP 4.3.0 */
$loc_de = setlocale(LC_ALL, 'de_DE@euro', 'de_DE', 'deu_deu');
echo "L'identifiant de l'allemand sur ce système est '$loc_de'";
?>
```

8.127.48 sha1_file() : Calcule le sha1 d'un fichier

string **sha1_file** (string filename , bool raw_output)

Calcule le sha1 du fichier filename en utilisant [US Secure Hash Algorithm 1](#) , puis retourne ce sha1. Le sha1 est un nombre hexadécimal de 40 caractères.

8.127.49 sha1() : Calcule le sha1 d'une chaîne de caractères

string **sha1** (string str , bool raw_output)

[sha1](#) calcule le sha1 de la chaîne de caractères str en utilisant [US Secure Hash Algorithm 1](#) , puis retourne cette valeur. Un sha1 est un nombre hexadécimal de 40 caractères. Si le paramètre optionnel raw_output est passé à TRUE , le sha1 est retourné sous forme binaire brute avec une taille de 20.

Note

Le paramètre optionnel raw_output a été ajouté en PHP 5.0.0 et vaut FALSE par défaut.

Exemple avec sha1

```
<?php
$str = 'pomme';

if (sha1($str) === '752c14ea195c460bac3c3b7896975ee9fd15eeb7') {
    echo "Voulez-vous une golden ou une spartan ?";
    exit;
}
?>
```

Voir aussi [sha1_file](#) , [crc32](#) et [md5](#)

8.127.50 similar_text() : Calcule la similarité de deux chaînes

int **similar_text** (string first , string second , float percent)

[similar_text](#) calcule la similarité entre les deux chaînes first et second , selon la méthode d'Oliver [1993]. Notez que cette implémentation n'utilise pas la méthode de pile comme dans le pseudo code d'Oliver, mais des appels récursifs, ce qui accélère ou pas le processus. Notez que la complexité de l'algorithme est en $O(N^3)$ où N est la taille de la plus grande chaîne.

Le troisième argument `percent` est une référence, où `similar_text` va calculer la similarité en pourcentage automatiquement. Sinon, elle retourne le nombre de caractères identiques dans les deux chaînes.

Voir aussi [levenshtein](#) et [soundex](#) .

8.127.51 `soundex()` : Calcule la clé soundex

string **soundex** (string `str`)

soundex calcule la clé soundex de la chaîne `str` .

La clé soundex possède la propriété qui fait que deux mots prononcés similairement auront la même clé soundex. Cette fonction est donc utilisée pour simplifier les recherches dans les bases de données, où vous connaissez la prononciation d'un mot ou nom, mais pas son orthographe exacte. Cette fonction retourne une chaîne de 4 caractères, commençant par une lettre.

L'implémentation de la fonction `soundex` de PHP a été décrite par Donald Knuth dans "The Art Of Computer Programming, vol. 3: Sorting And Searching", Addison-Wesley (1973), pp. 391-392 .

Notez bien que `soundex` prend en compte la prononciation à l'anglaise, et non pas à la française. Les résultats pourront donc être très différents de ce que l'on peut attendre. Des versions en langue française de `soundex` existent, mais elles ne sont pas intégrées dans PHP.

Exemples avec `soundex`

```
<?php
soundex("Euler")      == soundex("Ellery")    == 'E460';
soundex("Gauss")      == soundex("Ghosh")    == 'G200';
soundex("Hilbert")    == soundex("Heilbronn") == 'H416';
soundex("Knuth")      == soundex("Kant")     == 'K530';
soundex("Lloyd")      == soundex("Ladd")    == 'L300';
soundex("Lukasiewicz") == soundex("Lissajous") == 'L222';
?>
```

Voir aussi [levenshtein](#) , [metaphone](#) et [similar_text](#) .

8.127.52 `sprintf()` : Retourne une chaîne formatée

string **sprintf** (string `format` , *mixed args* , *mixed ...*)

sprintf retourne une chaîne formatée, avec le format `format` , en utilisant les arguments `args` .

La chaîne de format est composée de zéro, une ou plusieurs directives : les caractères ordinaires (à l'exception de %) qui sont copiés directement dans le résultat, et des **spécifications de conversion** , qui exploitent chacune un des arguments passés après la chaîne de format. Ces formats s'appliquent à `sprintf` et `printf` .

Chaque spécification de conversion est constituée d'un signe de pourcentage (%), suivi d'un ou plusieurs des éléments suivants, dans cet ordre :

1. Un **spécificateur d'espacement** optionnel qui dit quel spécificateur doit être utilisé pour espacer les résultats à la bonne taille de la chaîne. Cela peut être un caractère espace ou 0 (caractère zéro). Par défaut, l'espacement est effectué avec le caractère espace. Un

caractère d'espacement alternatif peut être spécifié en le préfixant avec un guillemet simple ('). Voir les exemples ci-dessous.

2. Un **remplisseur** optionnel qui indique quel caractère sera utilisé pour compléter le résultat jusqu'à la longueur requise. Ce peut être le caractère d'espace, ou le caractère 0 . Par défaut, le remplissage se fait avec des espaces. Un autre caractère de remplissage peut être spécifié en le préfixant avec un guillemet simple (') : voir les exemples ci-dessous.
3. Un **spécificateur d'alignement** qui indique si le résultat doit être aligné à gauche ou à droite. Par défaut, le résultat est aligné à gauche. Le caractère - fera que le résultat sera aligné à droite.
4. Un nombre optionnel, **spécificateur de taille** indique le nombre minimum de caractères que cette conversion doit fournir en résultat.
5. Un **spécificateur de précision** qui indique le nombre de décimales qui doivent être affichées pour les nombres à virgule flottante. Lorsque vous utilisez ce spécificateur dans une chaîne, il agit comme un point de coupure, définissant une limite maximale de caractères de la chaîne.
6. Un **spécificateur de type** qui indique le type avec lequel l'argument sera traité. Plusieurs types possibles :
 - ◆ % : un caractère de pourcentage littéral. Aucun argument n'est nécessaire.
 - ◆ b : l'argument est traité comme un entier, et présenté comme un nombre binaire.
 - ◆ c : l'argument est traité comme un entier, et présenté comme le caractère de code ASCII correspondant.
 - ◆ d : l'argument est traité comme un entier, et présenté comme un nombre décimal signé.
 - ◆ e : l'argument est traité comme une notation scientifique (e.g. 1.2e+2).
 - ◆ u : l'argument est traité comme un entier, et présenté comme un nombre décimal non signé.
 - ◆ f : l'argument est traité comme un nombre à virgule flottante (type float), et présenté comme un nombre à virgule flottante (tenant compte de la locale utilisée).
 - ◆ F : l'argument est traité comme un nombre à virgule flottante (type float), et présenté comme un nombre à virgule flottante (ne tenant pas compte de la locale utilisée). Disponible depuis PHP 4.3.10 et PHP 5.0.3.
 - ◆ o : l'argument est traité comme un entier, et présenté comme un nombre octal.
 - ◆ s : l'argument est traité et présenté comme une chaîne de caractères.
 - ◆ x : l'argument est traité comme un entier, et présenté comme un nombre hexadécimal (les lettres en minuscules).
 - ◆ X : l'argument est traité comme un entier, et présenté comme un nombre hexadécimal (les lettres en majuscules).

Depuis PHP 4.0.6, la chaîne de format supporte le numérotage et l'échange d'arguments. Par exemple :

Echange d'arguments

```
<?php
$format = "Il y a %d singes dans le %s";
printf($format, $num, $location);
?>
```

Ainsi, cet exemple peut afficher : "Il y a 5 singes dans le bananier". Mais imaginez que la chaîne de format soit créée dans un script séparé, comme une bibliothèque : cela arrive lorsqu'il faut internationaliser une application. Suivant la langue, il faudra peut-être écrire :

Echange d'arguments (2)

```
<?php
$format = "Le %s a %d singes";
printf($format, $num, $location);
?>
```

Ici, nous voyons bien le problème. L'ordre des arguments a été changé, et ne correspond plus à l'ordre des arguments dans le script PHP. Nous souhaitons laisser le code PHP intact, mais simplement indiquer dans la chaîne de formatage l'ordre dans lequel les arguments doivent être utilisés. La chaîne de format peut être réécrite ainsi :

Echange d'arguments (3)

```
<?php
$format = "Le %2\$s a %1\$d singes";
printf($format, $num, $location);
?>
```

Un des avantages est que vous pouvez désormais exploiter plusieurs fois les arguments sans les répéter. Ainsi :

Echange d'arguments (4)

```
<?php
$format = "Le %2\$s a %1\$d singes.
          C'est un beau %2\$s, avec %1\$d singes.";
printf($format, $num, $location);
?>
```

Voir aussi [printf](#) , [sscanf](#) , [fscanf](#) , [vsprintf](#) et [number_format](#) .

8.127.53 sscanf() : Analyse une chaîne à l'aide d'un format

mixed **sscanf** (string str , string format , *mixed* ...)

[sscanf](#) est l'inverse de la fonction [printf](#) . [sscanf](#) lit des données dans la chaîne str , et l'interprète en fonction du format format , qui est décrit dans la documentation de la fonction [sprintf](#) . Si seuls deux paramètres sont fournis, les valeurs trouvées seront retournées sous forme de tableau. Sinon, les valeurs seront placées dans un tableau. Si l'argument optionnel est passé, la fonction retournera le nombre de valeurs assignées. Le paramètre optionnel doit être passé par référence.

Tous les caractères blancs dans la chaîne format correspondent à un caractère blanc dans la chaîne str . Cela signifie que même une tabulation \t dans la chaîne de format peut correspondre à un simple espace dans la chaîne str .

Exemple avec [sscanf](#)

```
<?php
// Lecture d'un numéro de série
list($serial) = sscanf("SN/2350001", "SN/%d");
// et la date de fabrication
$mandate = "January 01 2000";
list($month, $day, $year) = sscanf($mandate, "%s %d %d");
echo "Le produit $serial a été fabriqué le : $year-" . substr($month, 0, 3) . "-$day\n";
?>
```

Si des paramètres optionnels sont passés, [sscanf](#) retournera le nombre de valeurs assignées.

Exemple avec [sscanf](#) : utilisation des paramètres optionnels

```
<?php
// lit les informations d'auteur, et génère une entrée DocBook
$auth = "24 Lewis Carroll";
$n = sscanf($auth, "%d\t%s %s", $id, $first, $last);
echo "<author id=\"\$id\">
      <firstname>$first</firstname>
      <surname>$last</surname>
</author>\n";
?>
```

Voir aussi [fscanf](#) , [printf](#) et [sprintf](#) .

8.127.54 [str_ireplace\(\)](#) : Version insensible à la casse de [str_replace](#)

mixed [str_ireplace](#) (mixed *search* , mixed *replace* , mixed *subject* , int *count*)

[str_ireplace](#) retourne une chaîne de caractères ou un tableau dont toutes les occurrences de *search* dans *subject* (en ignorant la casse), ont été remplacées par la valeur de *replace* . Si vous n'avez pas besoin de règles de recherche complexes, il est recommandé d'utiliser cette fonction qui est plus rapide que [eregi_replace](#) ou [preg_replace](#) , exploitant l'option *i* .

Si *subject* est un tableau, le remplacement est effectué sur chacun des éléments du sujet *subject* , et la valeur retournée est un aussi un tableau.

Si *search* et *replace* sont des tableaux, alors [str_replace](#) prendra une valeur de chaque tableau, et l'utilisera pour faire le remplacement dans *subject* . Si *replace* a moins de valeurs que *search* , alors une chaîne vide sera utilisée pour effectuer les remplacements. Si *search* est un tableau et que *replace* est une chaîne, alors la chaîne de remplacement sera utilisée pour chaque élément de *search* .

Exemple avec [str_ireplace](#)

```
<?php
$bodytag = str_ireplace("%body%", "black", "<body text=%BODY%>");
?>
```

Cette fonction est compatible avec les données binaires.

Note

Depuis PHP 5.0, le nombre de valeurs de *needle* trouvées et remplacées sera retourné dans le paramètre *count* passé par référence. Avant PHP 5.0.0, ce paramètre n'est pas disponible.

Voir aussi [str_replace](#) , [ereg_replace](#) , [preg_replace](#) et [strtr](#) .

8.127.55 [str_pad\(\)](#) : Complète une chaîne jusqu'à une taille donnée

string [str_pad](#) (string *input* , int *pad_length* , string *pad_string* , int *pad_type*)

[str_pad](#) retourne la chaîne *input* , complétée à droite, à gauche ou dans les deux sens, avec la chaîne *pad_string* jusqu'à ce qu'elle atteigne la taille de *pad_length* . Si le paramètre optionnel *pad_string* est omis, *input* sera complétée avec des espaces. Sinon, cette chaîne sera utilisée.

L'argument optionnel *pad_type* peut être l'une des constantes suivantes : `STR_PAD_RIGHT` , `STR_PAD_LEFT` , ou `STR_PAD_BOTH` . Si *pad_type* n'est pas spécifié, il prend la valeur par défaut de `STR_PAD_RIGHT` .

Si la valeur de *pad_length* est négative, ou inférieure à la taille courante de la chaîne *input* , *input* est retournée inchangée.

Exemple avec [str_pad](#)

```
<?php
$input = "Alien";
echo str_pad($input, 10); // affiche "Alien      "
echo str_pad($input, 10, "--", STR_PAD_LEFT); // affiche "--==Alien"
```

```

echo str_pad($input, 10, "_", STR_PAD_BOTH); // affiche "__Alien__"
echo str_pad($input, 6, "___");           // affiche "Alien_"
?>

```

Note

Le paramètre `pad_string` peut être tronqué si le nombre de caractères de complétion n'est pas multiple de la taille de `pad_string`.

8.127.56 str_repeat() : Répète une chaîne

string **str_repeat** (string input , int multiplier)

str_repeat retourne la chaîne `input_str` répétée `multiplier` fois. `multiplier` doit être positif ou nul. Si `multiplier` vaut 0, la fonction retourne la chaîne vide.

Exemple avec str_repeat

```

<?php
echo str_repeat("-", 10);
?>

```

L'exemple ci-dessus va afficher :

```
-----
```

Voir aussi [for](#) , [str_pad](#) et [substr_count](#) .

8.127.57 str_replace() : Remplace toutes les occurrences dans une chaîne

mixed **str_replace** (mixed search , mixed replace , mixed subject , int count)

str_replace retourne une chaîne ou un tableau, dont toutes les occurrences de `search` dans `subject` ont été remplacées par `replace` . Si vous n'avez pas besoin de règles de remplacement compliquées, utilisez cette fonction de préférence à [ereg_replace](#) et [preg_replace](#) .

Depuis PHP 4.0.5, chaque paramètre de **str_replace** peut être un tableau.

Attention

Avant PHP 4.3.3, un bogue existait lors de l'utilisation de tableaux avec les paramètres `search` et `replace` en même temps. Les index de `search` qui étaient vides étaient ignorés, mais le pointeur interne de `replace` n'était pas incrémenté. Cela a été corrigé en PHP 4.3.3, tout script s'appuyant sur ce bogue, doit supprimer les entrées vides avant d'appeler cette fonction pour imiter le comportement d'origine.

Si `subject` est un tableau, alors le remplacement se fera sur chaque élément de celui-ci, et la valeur retournée sera aussi un tableau.

Si `search` et `replace` sont des tableaux, alors **str_replace** prendra une valeur de chaque tableau, et l'utilisera pour faire le remplacement dans `subject` . Si `replace` a moins de valeurs que `search` , alors une chaîne vide sera utilisée pour effectuer les remplacements. Si `search` est un tableau et que `replace` est une chaîne, alors la chaîne de remplacement sera utilisée pour chaque élément de `search` . Cependant, l'inverse n'aurait aucun sens.

Exemple avec `str_replace`

```

<?php
// Génère : <body text='black'>
echo $bodytag = str_replace("%body%", "black", "<body text='%body%'>");

// Génère : Bnjr l mnd
$voyelles = array("a", "e", "i", "o", "u", "A", "E", "I", "O", "U");
echo $consonnes = str_replace($voyelles, "", "Bonjour le monde");

// Génère : Vous devriez manger des pizzas, des glaces et des gâteaux tous les jours.
$phrase = "Vous devriez manger des fruits, des légumes et des fibres tous les jours.";
$regime = array("fruits", "légumes", "fibres");
$bonne_chere = array("pizzas", "glaces", "gâteaux");

echo $newphrase = str_replace($regime, $bonne_chere, $phrase);

// Utilisation du compteur d'occurrences en PHP 5.0.0
$str = str_replace("a", "", "La disparition est un lipogramme en E.", $count);
echo $count; // 3
?>

```

Note

Cette fonction gère les chaînes binaires.

Note

Depuis PHP 5.0, le nombre de valeurs de search trouvées et remplacées seront retournées dans le paramètre count passé par référence. Avant PHP 5.0.0, ce paramètre n'est pas disponible.

Voir aussi [str_ireplace](#) , [substr_replace](#) , [ereg_replace](#) , [preg_replace](#) et [strtr](#) .

8.127.58 `str_rot13()` : Effectue une transformation ROT13

string `str_rot13` (string str)

`str_rot13` effectue un encodage ROT13 de la chaîne str et retourne le résultat. L'encodage ROT13 décale toutes les lettres de 13 dans l'alphabet, et laisse tous les autres caractères inchangés. L'encodage et le décodage est fait par la même fonction : passer le résultat de `str_rot13` à nouveau comme argument, retournera la chaîne originale.

Exemple avec `str_rot13`

```

<?php

echo str_rot13('PHP 4.3.0'); // CUC 4.3.0

?>

```

Note

Le comportement de cette fonction était bogué jusqu'en PHP 4.3.0. Avant cela, str était modifié, comme s'il était passé par référence.

8.127.59 `str_shuffle()` : Mélange les caractères d'une chaîne de caractères

string `str_shuffle` (string str)

`str_shuffle` mélange les caractères d'une chaîne. Une permutation parmi toutes celles possibles est créée.

Exemple `str_shuffle`

```
<?php
$str = 'abcdef';
$shuffled = str_shuffle($str);

// Cela va afficher une ligne comme : bfdaec
echo $shuffled;
?>
```

Voir aussi [shuffle](#) et [rand](#) .

8.127.60 `str_split()` : Convertit une chaîne de caractères en tableau

array **`str_split`** (string string , int *split_length*)

Convertit une chaîne de caractères en tableau. Si le paramètre optionnel `split_length` est spécifié, le tableau retourné sera découpé en sous-parties, chacune de taille `split_length` , sinon, chaque sous-partie aura la taille d'un caractère.

Cette fonction retourne FALSE si `split_length` est inférieur à 1. Si la longueur de `split_length` est supérieure à celle de string , la chaîne entière est retournée dans le premier (et seul) élément du tableau.

Exemples d'utilisation de `str_split`

```
<?php

$str = "Salut Dave";

$arr1 = str_split($str);
$arr2 = str_split($str, 3);

print_r($arr1);
print_r($arr2);

?>
```

L'exemple ci-dessus va afficher :

```
Array
(
[0] => S
[1] => a
[2] => l
[3] => u
[4] => t
[5] =>
[6] => D
[7] => a
[8] => v
[9] => e
)
```

```
Array
(
```

```
[0] => Sal
[1] => ut
[2] => Dav
[3] => e
)
```

Exemples liés à `str_split`

```
<?php
$str = "Salut Dave";

print $str{0}; // S
print $str{8}; // v

// Crée : array('S','a','l','u','t',' ','D','a','v','e')
$arr1 = preg_split('/\\/', $str, -1, PREG_SPLIT_NO_EMPTY);

?>
```

Voir aussi [chunk_split](#) , [preg_split](#) , [split](#) , [count_chars](#) , [str_word_count](#) et [for](#) .

8.127.61 `str_word_count()` : Compte le nombre de mots utilisés dans une chaîne

mixed `str_word_count` (string `string` , int `format` , string `charlist`)

`str_word_count` compte le nombre de mots dans la chaîne `string` . Si le paramètre optionnel `format` n'est pas spécifié, alors la valeur retournée sera un entier, représentant le nombre de mots trouvés. Si `format` est spécifié, la valeur retournée sera un tableau, qui dépend du format `format` . Les valeurs possibles pour `format` sont listées plus bas.

Dans cette fonction, la notion de "mot" dépend de la configuration de localisation. C'est une chaîne qui contient tous les caractères alphabétiques, et qui peut contenir, mais pas commencer par "" et "-".

8.127.62 `strcasecmp()` : Comparaison insensible à la casse de chaînes binaires

int `strcasecmp` (string `str1` , string `str2`)

`strcasecmp` retourne < 0 si `str1` est inférieure à `str2` ; > 0 si `str1` est plus grande que `str2` , et 0 si les deux chaînes sont égales.

Exemple avec `strcasecmp`

```
<?php
$var1 = "Hello";
$var2 = "hello";
if (strcasecmp($var1, $var2) == 0) {
    echo "$var1 est égale à $var2 (comparaison insensible à la casse)";
}
?>
```

Voir aussi [ereg](#) , [strcmp](#) , [substr](#) , [stristr](#) , [strncasecmp](#) et [strstr](#) .

8.127.63 `strchr()` : Alias de `strstr`

Cette fonction est un alias de : `strstr` .

8.127.64 `strcmp()` : Comparaison binaire de chaînes

int `strcmp` (string `str1` , string `str2`)

`strcmp` retourne < 0 si `str1` est inférieure à `str2` ; > 0 si `str1` est supérieure à `str2` , et 0 si les deux chaînes sont égales.

Notez que cette comparaison est sensible à la casse.

Voir aussi `ereg` , `strcasecmp` , `substr` , `stristr` , `strncasecmp` , `strncmp` et `strstr` .

8.127.65 `strcoll()` : Comparaison de chaînes localisées

int `strcoll` (string `str1` , string `str2`)

`strcoll` retourne < 0 si `str1` est inférieure à `str2` ; > 0 si `str1` est supérieure à `str2` , et 0 si les deux chaînes sont égales. Si la configuration de localisation courante est C ou POSIX, cette fonction est équivalente à `strcmp` .

Notez que cette comparaison est sensible à la casse, et que, contrairement à `strcmp` , elle n'est pas compatible avec les chaînes binaires.

Note
Le paramètre <code>strcoll</code> a été ajouté dans PHP 4.0.5 mais n'est activé dans la version win32 que depuis PHP 4.2.3.

Voir aussi `ereg` , `strcmp` , `strcasecmp` , `substr` , `stristr` , `strncasecmp` , `strncmp` , `strstr` et `setlocale` .

8.127.66 `strcspn()` : Trouve un segment de chaîne ne contenant pas certains caractères

int `strcspn` (string `str1` , string `str2` , int `start` , int `length`)

`strcspn` retourne la taille du premier segment de `str1` qui contient **aucun** des caractères de `str2` .

Depuis PHP 4.3.0, `strcspn` accepte deux paramètres de type `entier` optionnels qui peuvent être utilisés pour définir la position de départ `start` et la longueur `length` de la chaîne à examiner.

Note
Cette fonction gère les chaînes binaires.

Voir aussi `strspn` .

8.127.67 strip_tags() : Supprime les balises HTML et PHP d'une chaîne

string **strip_tags** (string str , string allowable_tags)

strip_tags retourne la chaîne str après avoir supprimé toutes les balises PHP et HTML du code. Elle génère des alertes si les balises sont incomplètes ou erronées. Elle utilise le même moteur de recherche que fgetss .

Le paramètre optionnel allowable_tags permet d'indiquer les balises qui doivent être conservées.

Note
allowable_tags a été ajouté en PHP 3.0.13 et PHP 4.0b3. Depuis PHP 4.3.0, les commentaires HTML sont aussi supprimés. Ce comportement est inclus en interne et ne peut pas être modifié avec le paramètre allowable_tags .
Attention
Comme <u>strip_tags</u> ne valide actuellement pas le HTML, les balises partielles ou rompues peuvent conduire à la suppression de plus de textes/données que désiré.
Attention
<u>strip_tags</u> ne modifie pas les attributs des balises que vous autorisez via le paramètre allowable_tags , y compris les attributs style et onmouseover , que des utilisateurs mal intentionnés peuvent utiliser.

Exemple avec <u>strip_tags</u>
<pre><?php \$text = '<p>Test paragraphe.</p> Autre texte'; echo strip_tags(\$text); echo "\n"; // Permet la balise <p> echo strip_tags(\$text, '<p>'); ?></pre>
L'exemple ci-dessus va afficher :
<pre>Test paragraphe. Autre texte <p>Test paragraphe</p>. Autre texte</pre>

strip_tags est protégé des données binaires depuis PHP 5.0.0.

Voir aussi htmlspecialchars .

8.127.68 stripslashes() : Décode une chaîne encodée avec addslashes

string **stripslashes** (string str)

stripslashes retourne la chaîne str après avoir supprimé tous les anti-slash. stripslashes respecte les séquences spéciales du C, telles que \n , \r ..., les nombres octaux et hexadécimaux.

Voir aussi addslashes .

8.127.69 stripos() : Recherche la première occurrence dans une chaîne, sans tenir compte de la casse

int **stripos** (string haystack , string needle , int offset)

Retourne la position numérique de la première occurrence de needle dans la chaîne haystack . Contrairement à strpos , stripos est insensible à la casse. Et, contrairement à strrpos , cette fonction peut prendre une chaîne complète comme paramètre needle et cette chaîne complète sera utilisée pour la recherche.

Si le paramètre needle n'est pas trouvé, stripos retournera FALSE .

Attention

Cette fonction peut retourner FALSE , mais elle peut aussi retourner une valeur équivalent à FALSE , utilisable dans une condition if simple. Utilisez l' opérateur === pour tester la valeur de retour exacte de cette fonction.

Exemples avec stripos

```
<?php
$findme    = 'a';
$mystring1 = 'xyz';
$mystring2 = 'ABC';

$pos1 = stripos($mystring1, $findme);
$pos2 = stripos($mystring2, $findme);

// Non, 'a' ne fait pas partie de 'xyz'
if ($pos1 === false) {
    echo "La chaîne '$findme' n'a pas été trouvée dans la chaîne '$mystring'";
}

// Notez l'utilisation de ===. Un simple == ne donnerait pas le résultat escompté
// car la lettre 'a' est à la position 0th (la première).
if ($pos2 !== false) {
    echo "La chaîne '$findme' a été trouvée dans la chaîne '$mystring'";
    echo " à la position $pos";
}
?>
```

Si needle n'est pas une chaîne, il est converti en entier, et utilisé comme caractère de code ASCII correspondant.

Le paramètre optionnel offset vous permet de spécifier à partir de quel caractère dans haystack vous souhaitez commencer la recherche. La position retournée sera toujours relative au début de la chaîne haystack .

Note

Cette fonction gère les chaînes binaires.

Voir aussi strpos , strrpos , strchr , substr , stristr , strstr , stripos et str_ireplace .

8.127.70 stripslashes() : Supprime les anti-slash d'une chaîne

string **stripslashes** (string str)

stripslashes retourne une chaîne dont les anti-slash ont été supprimés. \ devient ' , etc. Les doubles anti-slash sont réduits à un seul anti-slash. stripslashes est la fonction inverse de addslashes .

Note

Si magic_quotes_sybase est activée, aucun anti-slash n'est supprimé mais deux apostrophes sont remplacées par une seule à la place.

Un exemple d'utilisation de stripslashes est lorsque la directive PHP magic_quotes_gpc est à on (valeur par défaut) et que vous insérez des données dans une base de données qui requiert la protection des valeurs. Par exemple, si vous affichez simplement et directement des données provenant d'un formulaire HTML.

Exemple avec stripslashes

```
<?php
$str = "Avez-vous l\'oreille dure?";

// Affiche : Avez-vous l'oreille dure?
echo stripslashes($str);
?>
```

Note

stripslashes n'est pas récursif. Si vous voulez appliquer cette fonction à un tableau multi-dimensionnel, vous devez utiliser une fonction récursive.

Utilisation de stripslashes sur un tableau

```
<?php
function stripslashes_deep($value)
{
    $value = is_array($value) ?
        array_map('stripslashes_deep', $value) :
        stripslashes($value);

    return $value;
}

// Exemple
$array = array("f\\'oo", "b\\'ar", array("fo\\'o", "b\\'ar"));
$array = stripslashes_deep($array);

// Affichage
print_r($array);
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [0] => f'oo
    [1] => b'ar
    [2] => Array
        (
            [0] => fo'o
            [1] => b'ar
        )
)
```

Pour plus d'informations sur les "magic quotes", lisez la documentation sur [get_magic_quotes_gpc](#) .

Voir aussi [addslashes](#) et [get_magic_quotes_gpc](#) .

8.127.71 **stristr()** : Version insensible à la casse de **strstr**

string **stristr** (string haystack , string needle)

stristr retourne une sous-chaîne de haystack , allant de la première occurrence de needle jusqu'à la fin de la chaîne. needle et haystack sont traitées sans tenir compte de la casse.

Si needle est introuvable, **stristr** retourne FALSE .

Si needle n'est pas une chaîne, elle sera convertie en entier, et utilisée comme code ascii du caractère correspondant.

Exemple avec **stristr**

```
<?php
$email = 'USER@EXAMPLE.com';
echo stristr($email, 'e');
// Affiche ER@EXAMPLE.com
?>
```

Teste si une chaîne de caractères est trouvée ou pas

```
<?php
$string = 'Hello World!';
if(stristr($string, 'terre') === FALSE) {
    echo '"terre" non trouvé dans la chaîne de caractères';
}
// affiche : "terre" non trouvé dans la chaîne de caractères
?>
```

Utilisation d'une "chaîne de caractères" en chiffre

```
<?php
$string = 'POMME';
echo stristr($string, 112); // 112 = p minuscule
// affiche: POMME
?>
```

Note

Cette fonction gère les chaînes binaires.

Voir aussi [strstr](#) , [strchr](#) , [substr](#) et [ereg](#) .

8.127.72 **strlen()** : Calcule la taille d'une chaîne

int **strlen** (string string)

strlen retourne la taille de la chaîne string .

Exemple avec **strlen**

```
<?php
$str = 'abcdef';
```

```

echo strlen($str); // 6

$str = ' ab cd ';
echo strlen($str); // 7
?>

```

Voir aussi [count](#) et [mb_strlen](#) .

8.127.73 [strnatcasecmp\(\)](#) : Comparaison de chaînes avec l'algorithme d'"ordre naturel" (insensible à la casse)

```
int strnatcasecmp ( string str1 , string str2 )
```

[strnatcasecmp](#) implémente l'algorithme de comparaison qui ordonne les chaînes tel qu'un homme le ferait. Cette fonction est similaire à la fonction [strnatcmp](#) , mais la comparaison n'est pas sensible à la casse. Pour plus de détails, reportez-vous à [Natural Order String Comparison](#) de Martin Pool (en anglais).

[strnatcasecmp](#) retourne < 0 si str1 est inférieure à str2 ; > 0 si str1 est supérieure à str2 , et 0 si les deux chaînes sont égales.

Voir aussi [ereg](#) , [strcasecmp](#) , [substr](#) , [stristr](#) , [strcmp](#) , [strncmp](#) , [strncasecmp](#) , [strnatcmp](#) et [strstr](#) .

8.127.74 [strnatcmp\(\)](#) : Comparaison de chaînes avec l'algorithme d'"ordre naturel"

```
int strnatcmp ( string str1 , string str2 )
```

[strnatcmp](#) implémente l'algorithme de comparaison qui ordonne les chaînes tel qu'un homme le ferait. Cette fonction est similaire à la fonction [strnatcasecmp](#) , mais la comparaison n'est pas sensible à la casse. Un exemple de la différence de traitement avec l'algorithme standard est présenté ci-dessous :

Exemple avec [strnatcmp](#)

```

<?php
$arr1 = $arr2 = array("img12.png", "img10.png", "img2.png", "img1.png");
echo "Tri de chaînes standard\n";
usort($arr1, "strcmp");
print_r($arr1);
echo "\nTri de chaînes \"ordre naturel\"\n";
usort($arr2, "strnatcmp");
print_r($arr2);
?>

```

L'exemple ci-dessus va afficher :

```

Tri de chaînes standard
Array
(
    [0] => img1.png
    [1] => img10.png
    [2] => img12.png

```

```
[3] => img2.png
)
Tri de chaînes "ordre naturel"
Array
(
[0] => img1.png
[1] => img2.png
[2] => img10.png
[3] => img12.png
)
```

Pour plus de détails, reportez-vous à [Natural Order String Comparison](#) de Martin Pool (en anglais).

[strnatcmp](#) retourne < 0 si str1 est inférieure à str2 ; > 0 si str1 est supérieure à str2 , et 0 si les deux chaînes sont égales.

Notez que cette comparaison est sensible à la casse.

Voir aussi [ereg](#) , [strcasecmp](#) , [substr](#) , [stristr](#) , [strcmp](#) , [strncmp](#) , [strncasecmp](#) , [strnatcasecmp](#) , [strstr](#) , [natsort](#) et [natcasesort](#) .

8.127.75 [strncasecmp\(\)](#) : Compare en binaire des chaînes de caractères

int [strncasecmp](#) (string str1 , string str2 , int len)

[strncasecmp](#) est similaire à [strcasecmp](#) , à la différence près qu'elle permet de limiter le nombre de caractères utilisés pour comparer str1 et str2 , avec le paramètre len . Si une des chaînes est plus courte que len , alors la longueur de cette chaîne sera utilisée pour effectuer la comparaison.

[strncasecmp](#) retourne < 0 si str1 est plus petite que str2 ; > 0 si str1 est plus grande que str2 , et 0 si elles sont égales.

Voir aussi [ereg](#) , [strcasecmp](#) , [strcmp](#) , [substr](#) , [stristr](#) et [strstr](#) .

8.127.76 [strncmp\(\)](#) : Comparaison binaire des n premiers caractères

int [strncmp](#) (string str1 , string str2 , int len)

[strncmp](#) est identique à la fonction [strcmp](#) , avec la différence que vous pouvez spécifier le nombre maximum de caractères à utiliser pour la comparaison de str1 avec str2 grâce au paramètre len .

[strncmp](#) retourne < 0 si str1 est inférieure à str2 ; > 0 si str1 est supérieure à str2 , et 0 si les deux chaînes sont égales.

Notez que cette comparaison est sensible à la casse.

Voir aussi [ereg](#) , [strncasecmp](#) , [strcasecmp](#) , [substr](#) , [stristr](#) , [strcmp](#) et [strstr](#) .

8.127.77 `strpbrk()` : Recherche une chaîne de caractères dans un ensemble de caractères

string **strpbrk** (string haystack , string char_list)

`strpbrk` recherche la chaîne haystack dans l'ensemble de caractères char_list et retourne une chaîne commençant au caractère trouvé (ou FALSE s'il n'est pas trouvé).

Note

Le paramètre char_list est sensible à la casse.

Exemple avec `strpbrk`

```
<?php
$text = 'Ceci est un texte Simple.';

// Ceci affichera "i est un texte simple." car 'i' correspond en premier
echo strpbrk($text, 'mi');

// Ceci affichera "Simple." car les caractères sont sensibles à la casse
echo strpbrk($text, 'S');
?>
```

8.127.78 `strpos()` : Trouve la position d'un caractère dans une chaîne

int **strpos** (string haystack , mixed needle , int offset)

`strpos` retourne la position numérique de la première occurrence de needle dans la chaîne de caractères haystack . Contrairement à la fonction `strpos` , celle-ci peut prendre une chaîne de caractères complète comme paramètre needle et cette chaîne sera utilisée en totalité.

Si needle n'est pas trouvée, la fonction retourne FALSE .

Attention

Cette fonction peut retourner FALSE , mais elle peut aussi retourner une valeur équivalent à FALSE , utilisable dans une condition if simple. Utilisez l' opérateur === pour tester la valeur de retour exacte de cette fonction.

Note

Cette fonction gère les chaînes binaires.

Exemple avec `strpos`

```
<?php
$string = 'abc';
$findme = 'a';
$pos = strpos($string, $findme);

// Notez l'utilisation de ===. Un simple == ne donnerait pas le résultat escompté
// car la lettre 'a' est à la position 0 (la première).
if ($pos === false) {
    echo "La chaîne '$findme' n'a pas été trouvée dans la chaîne '$string'";
} else {
    echo "La chaîne '$findme' a été trouvée dans la chaîne '$string'";
    echo " et à la position $pos";
}
```

```
// On peut chercher le caractère, en ignorant tout avant une position
$newstring = 'abcdef abcdef';
$pos = strpos($newstring, 'a', 1); // $pos = 7, pas 0
?>
```

Si needle n'est pas une chaîne, il est converti en entier, et utilisé comme caractère de code ASCII correspondant.

Le paramètre optionnel offset vous permet de spécifier à partir de quel caractère dans haystack vous souhaitez commencer la recherche. La position retournée sera toujours relative au début de la chaîne haystack .

Voir aussi [strrpos](#) , [stripos](#) , [stripos](#) , [strrchr](#) , [substr](#) , [stristr](#) et [strstr](#) .

8.127.79 strrchr() : Trouve la dernière occurrence d'un caractère dans une chaîne

string **strrchr** (string haystack , string needle)

[strrchr](#) retourne le segment de la chaîne haystack qui commence avec la dernière occurrence de needle , jusqu'à la fin de la chaîne haystack .

[strrchr](#) retourne FALSE si needle n'est pas trouvé.

Si needle contient plus d'un caractère, seul le premier sera utilisé en PHP 4. Ce comportement est différent de celui de [strchr](#)

Si needle n'est pas une chaîne, il est converti en entier, et utilisée comme caractère de code ASCII correspondant.

Exemple avec [strrchr](#)

```
<?php
// Lit le dernier dossier dans le chemin $PATH
$dir = substr(strrchr($PATH, ":"), 1);

// Lit tout, après la dernière nouvelle ligne
$text = "Ligne 1\nLigne 2\nLigne 3";
$last = substr(strrchr($text, 10), 1 );
?>
```

[strrchr](#) a été protégée des données binaires depuis PHP 4.3.0.

Voir aussi [strstr](#) , [substr](#) , [stristr](#) et [strstr](#) .

8.127.80 strrev() : Inverse une chaîne

string **strrev** (string string)

strrev retourne la chaîne string , après avoir changé l'ordre des caractères.

Inverser une chaîne avec [strrev](#)

```
<?php
echo strrev("Bonjour le monde!");
```

```
// affiche "ednom el ruojnoB"
?>
```

8.127.81 **stripos()** : Trouve la position de la dernière occurrence d'une chaîne dans une autre de façon insensible à la casse

int **stripos** (string haystack , string needle , int offset)

Retourne la position numérique de la dernière occurrence de needle dans la chaîne de caractères haystack . Contrairement à strrpos , stripos est insensible à la casse. Notez aussi que les positions dans les chaînes commencent à 0, et non à 1.

Notez que needle peut être une chaîne constituée d'un ou de plusieurs caractères.

Si needle n'est pas trouvé, la fonction retourne FALSE .

Attention

Cette fonction peut retourner FALSE , mais elle peut aussi retourner une valeur équivalent à FALSE , utilisable dans une condition if simple. Utilisez l' opérateur === pour tester la valeur de retour exacte de cette fonction.

Un exemple simple avec stripos

```
<?php
$haystack = 'ababcd';
$needle   = 'aB';

$pos      = stripos($haystack, $needle);

if ($pos === false) {
    echo "Désolé, impossible de trouver ($needle) dans ($haystack)";
} else {
    echo "Félicitations !\n";
    echo "Nous avons trouvé le dernier ($needle) dans ($haystack) à la position ($pos)";
}
?>
```

L'exemple ci-dessus va afficher :

Félicitations !

Nous avons trouvé le dernier (aB) dans (ababcd) à la position (2)

Le paramètre offset peu être spécifié pour commencer à chercher un nombre arbitraire de caractères dans la chaîne.

Les valeurs négatives commenceront la recherche aux caractères offset à partir du **début** de la chaîne de caractères.

Voir aussi strrpos , strchr , substr , stripos et stristr .

8.127.82 **strrpos()** : Trouve la position de la dernière occurrence d'un caractère dans une chaîne

int **strrpos** (string haystack , string needle , int offset)

strrpos retourne la position numérique de la dernière occurrence du caractère needle dans la chaîne haystack . Notez que needle doit être un caractère unique en PHP 4. Si une chaîne est passée dans ce paramètre, seul le premier caractère sera utilisé.

Si needle n'est pas trouvé, **strrpos** retournera FALSE .

Il est facile de confondre le résultat "caractère trouvé à la position 0" et "caractère non trouvé". Voici comment faire la différence :

Exemple avec **strrpos**

```
<?php
// A partir de PHP 4.0b3 :
$pos = strrpos($machaine, "b");
if ($pos === false) { // note : trois signes égal
    // pas trouvé
}

// en versions plus anciennes que 4.0b3 :
$pos = strrpos($machaine, "b");
if (is_bool($pos) && !$pos) {
    // pas trouvé
}
?>
```

Si needle n'est pas une chaîne, il est converti en entier, et utilisé comme caractère de code ASCII correspondant.

Note

Depuis PHP 5.0.0, offset peut être spécifié pour commencer à rechercher un nombre de caractères arbitraires dans la chaîne. Les valeurs négatives vont stopper la recherche à partir d'un point arbitraire avant la fin de la chaîne.

Note

Le paramètre needle peut être composé de plus d'un caractère à partir de PHP 5.0.0.

Voir aussi [strpos](#) , [stripos](#) , [strchr](#) , [substr](#) , [stristr](#) et [strstr](#) .

8.127.83 **strspn()** : Trouve le premier segment de chaîne

int **strspn** (string str1 , string str2 , int start , int length)

strspn retourne la taille du segment initial de la chaîne str1 qui est entièrement constitué de caractères contenus dans str2 .

Exemple avec **strspn**

```
<?php
echo strspn("42 est la réponse, mais quelle est la question ...", "1234567890");
?>
```

va afficher 3, car la chaîne "42" est le plus long segment initial de "42 est la réponse, mais quelle est la question ...", dont tous les caractères sont contenus dans "1234567890".

De plus, elle accepte deux paramètres optionnels entiers qui peuvent être utilisés pour définir la position de départ et la longueur de la chaîne à examiner.

Depuis PHP 4.3.0, [strspn](#) accepte deux paramètres de type [entier](#) optionnels qui peuvent être utilisés pour définir la position start et la longueur length de la chaîne à examiner.

Exemple avec les paramètres optionnels
<pre><?php echo strspn("foo", "o", 1, 2); // 2 ?></pre>
Note
Cette fonction gère les chaînes binaires.

Voir aussi [strcspn](#) .

8.127.84 strstr() : Trouve la première occurrence dans une chaîne

string **strstr** (string haystack , string needle)

[strstr](#) retourne une sous-chaîne de haystack , allant de la première occurrence de needle jusqu'à la fin de la chaîne.

Si needle est introuvable, [strstr](#) retourne FALSE .

Si needle n'est pas une chaîne, elle sera convertie en entier, et utilisée comme code ascii du caractère correspondant.

Note
strstr est sensible à la casse. Pour une fonctionnalité identique, mais insensible à la casse, reportez-vous à stristr .

Exemple avec strstr
<pre><?php \$email = 'user@example.com'; \$domaine = strstr(\$email, '@'); echo \$domaine; // @example.com ?></pre>

Note
Si vous voulez juste déterminer si un needle particulier se trouve dans la chaîne haystack , utilisez la fonction strpos qui est plus rapide et qui consomme moins de ressources.

[strstr](#) a été protégée des données binaires depuis PHP 4.3.0.

Voir aussi [ereg](#) , [preg_match](#) , [stristr](#) , [strpos](#) , [strchr](#) et [substr](#) .

8.127.85 strtok() : Coupe une chaîne en segments

string **strtok** (string str , string token)

[strtok](#) coupe la chaîne str en segment, chaque segment étant délimité par token . Par exemple, si vous avez une chaîne telle que "Voici un bon exemple" , vous pouvez en extraire les différents mots en utilisant cette fonction :

Exemple avec `strtok`

```
<?php
$string = "Ceci est\tun bon\nexemple";
// Utilisez aussi les nouvelles lignes et les tabulations
// comme séparateur de mots
$tok = strtok($string, " \n\t");

while ($tok !== false) {
    echo "Mot = $tok<br />";
    $tok = strtok(" \n\t");
}
?>
```

Notez que seul le premier appel à `strtok` nécessite les deux arguments. Tous les appels ultérieurs à `strtok` ne nécessitent que le délimiteur. Pour initialiser à nouveau `strtok`, ou pour recommencer, fournissez à nouveau le paramètre `str`. La chaîne `str` sera découpée dès que l'un des caractères de token est trouvé.

Le comportement de cette fonction avec la chaîne vide a changé depuis PHP 4.1.0. L'ancien comportement était de retourner une chaîne vide, tandis que le nouveau comportement, plus correct, retourne `FALSE`

Ancien comportement de `strtok`

```
<?php
$first_token = strtok('/chose', '/');
$second_token = strtok('/');
var_dump($first_token, $second_token);
?>
```

L'exemple ci-dessus va afficher :

```
string(0) ""
string(9) "chose"
```

Nouveau comportement de `strtok`

```
<?php
$first_token = strtok('/chose', '/');
$second_token = strtok('/');
var_dump($first_token, $second_token);
?>
```

L'exemple ci-dessus va afficher :

```
string(9) "chose"
bool(false)
```

Attention

Cette fonction peut retourner `FALSE`, mais elle peut aussi retourner une valeur équivalent à `FALSE`, utilisable dans une condition `if` simple. Utilisez l'opérateur `===` pour tester la valeur de retour exacte de cette fonction.

Voir aussi [split](#) et [explode](#) .

8.127.86 strtolower() : Renvoie une chaîne en minuscules

string **strtolower** (string str)

[strtolower](#) retourne string , après avoir converti tous les caractères alphabétiques en minuscules.

Notez que la notion d'"alphabétique" est déterminée par la configuration de localisation. Cela signifie que pour la configuration par défaut "C", les caractères tels que les voyelles accentuées (comme é, è ou à) ne seront pas convertis.

Exemple avec [strtolower](#)

```
<?php
$str = "Marie A un Petit Agneau, et l'aime TRÈS fORT.";
$str = strtolower($str);
echo $str; // marie a un petit agneau, et l'aime très fort.
?>
```

Note

Cette fonction gère les chaînes binaires.

Voir aussi [strtoupper](#) , [ucfirst](#) , [ucwords](#) et [mb strtolower](#) .

8.127.87 strtoupper() : Renvoie une chaîne en majuscules

string **strtoupper** (string string)

[strtoupper](#) retourne string , après avoir converti tous les caractères alphabétiques en majuscules.

Notez que la notion d'"alphabétique" est déterminée par la configuration de localisation. Cela signifie que pour la configuration par défaut "C", les caractères tels que les voyelles accentuées (comme é, è ou à) ne seront pas converties.

Exemple avec [strtoupper](#)

```
<?php
$str = "Marie A un Petit Agneau, et l'aime fORT.";
$str = strtoupper($str);
echo $str; // MARIE A UN PETIT AGNEAU, ET L'AIME FORT.

// Note : Très aurait été converti en TRÈS
?>
```

Note

Cette fonction gère les chaînes binaires.

Voir aussi [strtolower](#) , [ucfirst](#) , [ucwords](#) et [mb strtoupper](#) .

8.127.88 strstr() : Remplace des caractères dans une chaîne

string **strstr** (string str , string from , string to) string **strstr** (string str , array replace_pairs)

strstr retourne la chaîne str , après avoir remplacé chaque caractère du paramètre from par son équivalent dans le paramètre to .

Si from et to sont de tailles différentes, les caractères en trop dans l'un ou l'autre seront ignorés.

Exemple avec strstr

```
<?php
$addr = strstr($addr, "ääö", "ao");
?>
```

strstr peut être appelée avec uniquement deux arguments. Dans ce cas, elle se comporte d'une autre façon : from doit alors être un tableau associatif de paires, qui seront remplacées dans la chaîne str . **strstr** commence toujours par rechercher la chaîne la plus longue, et ne travaille pas sur des segments qu'elle a déjà modifié.

Exemple de strstr avec deux arguments

```
<?php
$trans = array("bonjour" => "salut", "salut" => "bonjour");
echo strstr("salut à tous, j'ai dit bonjour", $trans);
?>
```

L'exemple ci-dessus va afficher :

bonjour à tous, j'ai dit salut

Note

Les paramètres optionnels to et from ont été ajoutés en PHP 4.0.0.

Voir aussi [ereg_replace](#) .

8.127.89 substr_compare() : Compare deux chaînes depuis un offset jusqu'à une longueur en caractères

int **substr_compare** (string main_str , string str , int offset , int length , bool case_insensitivity)

substr_compare compare main_str à partir de la position offset avec str pendant length caractères.

Cette fonction retourne < 0 si main_str à partir de offset est inférieur à str , > 0 s'il est plus grand que str , et 0 si ils sont égaux. Si length est égal ou plus grand que la taille de main_str et que length est configuré, **substr_compare** affiche une alerte et retourne FALSE .

Si case_insensitivity est à TRUE , la comparaison n'est pas sensible à la casse.

Exemple avec substr_compare

```
<?php
echo substr_compare("abcde", "bc", 1, 2); // 0
echo substr_compare("abcde", "bcg", 1, 2); // 0
echo substr_compare("abcde", "BC", 1, 2, true); // 0
echo substr_compare("abcde", "bc", 1, 3); // 1
```

```

echo substr_compare("abcde", "cd", 1, 2); // -1
echo substr_compare("abcde", "abc", 5, 1); // warning
?>

```

8.127.90 substr_count() : Compte le nombre d'occurrences de segments dans une chaîne

int **substr_count** (string haystack , string needle , int offset , int length)

substr_count retourne le nombre d'occurrences de needle dans la chaîne haystack . Notez que needle est sensible à la casse.

Note

Cette fonction ne compte pas les chaînes de caractères qui se recouvrent. Voyez l'exemple ci-dessous !

8.127.91 substr_replace() : Remplace un segment dans une chaîne

mixed **substr_replace** (mixed string , string replacement , int start , int length)

substr_replace remplace un segment de la chaîne string par la chaîne replacement . Le segment est délimité par start et éventuellement par length . La chaîne de résultat est retournée. Si string est un tableau, alors un tableau est retourné.

Si start est positif, le remplacement se fera à partir du start -ième caractère de string .

Si start est négatif, le remplacement se fera à partir du start -ième caractère en partant de la fin de la chaîne string .

Si length est fourni et positif, il représentera la longueur du segment de code remplacé dans la chaîne string . S'il est négatif, il représentera la longueur du segment remplacé, mais compté dans l'ordre inverse de la chaîne string . S'il est omis, il prendra la valeur par défaut de la taille de la chaîne, et remplacera tout jusqu'à la fin de la chaîne string .

Exemple avec substr_replace

```

<?php
$var = 'ABCDEFGH:/MNRPQR/';
echo "Original : $var<hr>\n";

// Remplace toute la chaîne $var par 'bob'.
echo substr_replace($var, 'bob', 0) . "<br />\n";
echo substr_replace($var, 'bob', 0, strlen($var)) . "<br />\n";

// Insert 'bob' au début de la chaîne
echo substr_replace($var, 'bob', 0, 0) . "<br />\n";

// Remplace la séquence 'MNRPQR' par 'bob'.
echo substr_replace($var, 'bob', 10, -1) . "<br />\n";
echo substr_replace($var, 'bob', -7, -1) . "<br />\n";

// Efface la séquence 'MNRPQR' de $var.
echo substr_replace($var, '', 10, -1) . "<br />\n";
?>

```

Note

Cette fonction gère les chaînes binaires.

Voir aussi [str_replace](#) et [substr](#) .

8.127.92 substr() : Retourne un segment de chaîne

string **substr** (string string , int start , int length)

[substr](#) retourne le segment de string défini par start et length .

Si start est positif, la chaîne retournée commencera au caractère numéro start , dans la chaîne string . Le premier caractère est numéroté zéro.

Exemple avec [substr](#)

```
<?php
echo substr('abcdef', 1);      // bcdef
echo substr('abcdef', 1, 3);  // bcd
echo substr('abcdef', 0, 4);  // abcd
echo substr('abcdef', 0, 8);  // abcdef
echo substr('abcdef', -1, 1); // f

// Accéder à un simple caractère dans une chaîne
// peut également être réalisé en utilisant des accolades
$string = 'abcdef';
echo $string{0};              // a
echo $string{3};              // d
echo $string{strlen($string)-1}; // f

?>
```

Si start est négatif, la chaîne retournée commencera au caractère numéro start à compter de la fin de la chaîne string .

Exemple de start négatif

```
<?php
$rest = substr("abcdef", -1); // retourne "f"
$rest = substr("abcdef", -2); // retourne "ef"
$rest = substr("abcdef", -3, 1); // retourne "d"
?>
```

Si length est fourni et est positif, la chaîne retournée contiendra au plus length caractères, en commençant à partir du caractère start (en fonction de la taille de la chaîne string). Si string est plus petite que start , [substr](#) retournera FALSE .

Si length est fourni et négatif, alors le même nombre de caractères sera omis, en partant de la fin de la chaîne string . Si start représente une position hors de la chaîne, une chaîne vide sera retournée.

Utiliser une valeur négative avec length

```
<?php
$rest = substr("abcdef", 0, -1); // retourne "abcde"
$rest = substr("abcdef", 2, -1); // retourne "cde"
$rest = substr("abcdef", 4, -4); // retourne ""
$rest = substr("abcdef", -3, -1); // retourne "de"
?>
```

Voir aussi [strrchr](#) , [substr_replace](#) , [ereg](#) , [trim](#) , [mb_substr](#) et [wordwrap](#) .

8.127.93 trim() : Supprime les espaces (ou d'autres caractères) en début et fin de chaîne

string **trim** (string *str* , *string charlist*)

trim retourne la chaîne *str* , après avoir supprimé les caractères invisibles en début et fin de chaîne. Si le second paramètre *charlist* est omis, **trim** supprimera les caractères suivants :

- " " (ASCII 32 (0x20)), un espace ordinaire.
- "\t" (ASCII 9 (0x09)), une tabulation.
- "\n" (ASCII 10 (0x0A)), une nouvelle ligne (line feed).
- "\r" (ASCII 13 (0x0D)), un retour chariot (carriage return).
- "\0" (ASCII 0 (0x00)), le caractère NUL .
- "\x0B" (ASCII 11 (0x0B)), une tabulation verticale.

8.127.94 ucfirst() : Met le premier caractère en majuscule

string **ucfirst** (string *str*)

ucfirst retourne la chaîne *str* après avoir remplacé le premier caractère par sa majuscule, si le premier caractère est alphabétique.

Notez que la notion d'"alphabétique" est déterminée par la configuration de localisation. Cela signifie que pour la configuration par défaut "C", les caractères tels que les voyelles accentuées (comme é , è ou à) ne seront pas converties.

Exemple avec **ucfirst**

```
<?php
$foo = 'bonjour tout le monde!';
$foo = ucfirst($foo);           // Bonjour tout le monde!

$bar = 'BONJOUR TOUT LE MONDE!';
$bar = ucfirst($bar);         // BONJOUR TOUT LE MONDE!
$bar = ucfirst(strtolower($bar)); // Bonjour tout le monde!
?>
```

Voir aussi [strtolower](#) , [strtoupper](#) et [ucwords](#) .

8.127.95 ucwords() : Met en majuscule la première lettre de tous les mots

string **ucwords** (string *str*)

ucwords retourne la chaîne *str* après avoir mis en majuscule la première lettre de tous les mots, si ce caractère est alphabétique.

La définition d'un mot est : toute séquence de caractères qui suit immédiatement un caractère blanc (espace, tabulation, nouvelle ligne, retour chariot, tabulation verticale).

Exemple avec **ucwords**

```
<?php
$foo = 'bonjour tout le monde!';
$foo = ucwords($foo);         // Bonjour Tout Le Monde!
```

```
$bar = 'BONJOUR TOUT LE MONDE!';
$bar = ucwords($bar);           // BONJOUR TOUT LE MONDE!
$bar = ucwords(strtolower($bar)); // Bonjour Tout Le Monde!
?>
```

Note

Cette fonction gère les chaînes binaires.

Voir aussi [strtoupper](#) , [strtolower](#) et [ucfirst](#) .

8.127.96 vfprintf() : Ecrit une chaîne formatée dans un flux

int **vfprintf** (resource handle , string format , array args)

[vfprintf](#) écrit une chaîne produite en accord avec le paramètre format dans le flux handle . format est décrit dans la page de documentation de la fonction [sprintf](#) .

Agit de la même façon que [fprintf](#) excepté que [vfprintf](#) accepte un tableau d'arguments, plutôt qu'un nombre variant d'arguments.

Retourne la longueur de la chaîne produite.

Voir aussi [printf](#) , [sprintf](#) , [scanf](#) , [fscanf](#) , [vsprintf](#) et [number_format](#) .

8.127.97 vprintf() : Affiche une chaîne formatée

int **vprintf** (string format , array args)

[vprintf](#) affiche le tableau args , sous forme de chaîne formatée grâce à format Le format est le même que celui utilisé par [sprintf](#) .

[vprintf](#) fonctionne comme [printf](#) mais accepte un tableau comme argument, au lieu d'une liste d'arguments.

Retourne la longueur de la chaîne affiché.

Voir aussi [printf](#) , [sprintf](#) et [vsprintf](#) .

8.127.98 vsprintf() : Retourne une chaîne formatée

string **vsprintf** (string format , array args)

[vsprintf](#) retourne une chaîne formatée à partir du tableau de valeurs args , et en utilisant le format format . Le format est le même que celui utilisé par [sprintf](#) .

[vsprintf](#) fonctionne comme [sprintf](#) mais accepte un tableau comme argument, au lieu d'une liste d'arguments.

Voir aussi [sprintf](#) et [vprintf](#) .

8.127.99 wordwrap() : Effectue la césure d'une chaîne

string **wordwrap** (string *str* , int *width* , string *break* , bool *cut*)

wordwrap retourne la chaîne *str* , après avoir inséré *break* tous les *width* caractères.

Par défaut, wordwrap va automatiquement insérer une nouvelle ligne en utilisant `\n` tous les 75 caractères, si *width* ou *break* ne sont pas fournis.

Si le paramètre *cut* est mis à 1, la césure de la chaîne sera toujours à la taille *width* . Si vous avez un mot qui est plus long que la taille de césure, il sera coupé en morceaux. (Voir le second exemple.)

Note

Le paramètre optionnel *cut* a été ajouté en PHP 4.0.3.

Exemple avec wordwrap

```
<?php
$text = "Portez ce vieux whisky au juge blond qui fume.";
$newtext = wordwrap( $text, 20 );

echo "$newtext\n";
?>
```

L'exemple ci-dessus va afficher :

```
Portez ce vieux
whisky au juge
blond qui fume.
```

Exemple avec wordwrap avec césure

```
<?php
$text = "Un mot très très loooooooooooooooooooooong.";
$newtext = wordwrap($text, 8, "\n", 1);

echo "$newtext\n";
?>
```

L'exemple ci-dessus va afficher :

```
Un mot
très
très
looooooo
oooooo
ooong.
```

Voir aussi [nl2br](#) et [chunk_split](#) .

8.128 Shockwave Flash

8.128.1 Introduction

PHP a la capacité de créer des animations Shockwave Flash grâce au module de Paul Haeberli : libswf module.

Note

Le support de SWF a été ajouté en PHP 4 RC2.

La bibliothèque libswf ne propose pas de support pour Windows. Le développement de cette bibliothèque a été stoppé, et la source n'est pas disponible pour assurer le portage vers les autres systèmes.

Pour une version plus à jour d'extension SWF, regardez donc les fonctions MING .

8.128.2 Pré-requis

Vous avez besoin de la bibliothèque libswf pour compiler PHP avec le support de cette extension. Vous pouvez télécharger cette bibliothèque <ftp://ftp.sgi.com/cgi/graphics/grafica/flash/> .

8.128.3 Installation

Une fois que vous avez libswf, tout ce qui reste à faire est de configurer PHP avec --with-swf[=DIR] où DIR est le dossier qui accueille les dossiers de include et lib. Le dossier include doit contenir le fichier swf.h file et le dossier lib doit contenir le fichier libswf.a . Si vous décompressez la distribution de libswf, les deux fichiers seront dans le même dossier. Par conséquent, vous devrez les mettre dans le dossier ad hoc manuellement.

8.128.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.128.5 Types de ressources

Cette extension ne définit aucune ressource.

8.128.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

MOD_COLOR (entier)
MOD_MATRIX (entier)
TYPE_PUSHBUTTON (entier)
TYPE_MENUBUTTON (entier)
BShitTest (float)

BSDown (float)
 BSOver (float)
 BSUp (float)
 OverDowntoldle (entier)
 IdletoOverDown (entier)
 OutDowntoldle (entier)
 OutDowntoOverDown (entier)
 OverDowntoOutDown (entier)
 OverUptoOverDown (entier)
 OverUptoldle (entier)
 IdletoOverUp (entier)
 ButtonEnter (entier)
 ButtonExit (entier)
 MenuEnter (entier)
 MenuExit (entier)

8.128.7 Exemples

Une fois que vous avez réussi à installer PHP avec Shockwave Flash, vous pouvez créer des animations Flash avec PHP. Vous serez surpris du résultat. Essayez donc ceci :

Exemple SWF

```

<?php
  swf_openfile ("test.swf", 256, 256, 30, 1, 1, 1);
  swf_ortho2 (-100, 100, -100, 100);
  swf_defineline (1, -70, 0, 70, 0, .2);
  swf_definerect (4, 60, -10, 70, 0, 0);
  swf_definerect (5, -60, 0, -70, 10, 0);
  swf_addcolor (0, 0, 0, 0);

  swf_definefont (10, "Mod");
  swf_fontsize (5);
  swf_fontslant (10);
  swf_definetext (11, "This be Flash wit PHP!", 1);

  swf_pushmatrix ();
  swf_translate (-50, 80, 0);
  swf_placeobject (11, 60);
  swf_popmatrix ();

  for ($i = 0; $i < 30; $i++) {
    $p = $i/(30-1);
    swf_pushmatrix ();
    swf_scale (1-($p*.9), 1, 1);
    swf_rotate (60*$p, 'z');
    swf_translate (20+20*$p, $p/1.5, 0);
    swf_rotate (270*$p, 'z');
    swf_addcolor ($p, 0, $p/1.2, -$p);
    swf_placeobject (1, 50);
    swf_placeobject (4, 50);
    swf_placeobject (5, 50);
    swf_popmatrix ();
    swf_showframe ();
  }

  for ($i = 0; $i < 30; $i++) {
    swf_removeobject (50);
    if (($i%4) == 0) {
      swf_showframe ();
    }
  }

```

```
}  
  
swf_startdoaction ();  
swf_actionstop ();  
swf_enddoaction ();  
  
swf_closefile ();  
?>
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [swf_actiongeturl](#)
- [swf_actiongotoframe](#)
- [swf_actiongotolabel](#)
- [swf_actionnextframe](#)
- [swf_actionplay](#)
- [swf_actionprevframe](#)
- [swf_actionsettarget](#)
- [swf_actionstop](#)
- [swf_actiontogglequality](#)
- [swf_actionwaitforframe](#)
- [swf_addbuttonrecord](#)
- [swf_addcolor](#)
- [swf_closefile](#)
- [swf_definebitmap](#)
- [swf_definefont](#)
- [swf_defineline](#)
- [swf_definepoly](#)
- [swf_definerect](#)
- [swf_definetext](#)
- [swf_endbutton](#)
- [swf_enddoaction](#)
- [swf_endshape](#)
- [swf_endsymbol](#)
- [swf_fontsize](#)
- [swf_fontslant](#)
- [swf_fontracking](#)
- [swf_getbitmapinfo](#)
- [swf_getfontinfo](#)
- [swf_getframe](#)
- [swf_labelframe](#)
- [swf_lookat](#)
- [swf_modifyobject](#)
- [swf_mulcolor](#)
- [swf_nextid](#)
- [swf_oncondition](#)
- [swf_openfile](#)
- [swf_ortho](#)
- [swf_ortho2](#)
- [swf_perspective](#)

- [swf_placeobject](#)
- [swf_polarview](#)
- [swf_popmatrix](#)
- [swf_posround](#)
- [swf_pushmatrix](#)
- [swf_removeobject](#)
- [swf_rotate](#)
- [swf_scale](#)
- [swf_setfont](#)
- [swf_setframe](#)
- [swf_shapearc](#)
- [swf_shapecurveto](#)
- [swf_shapecurveto3](#)
- [swf_shapefillbitmapclip](#)
- [swf_shapefillbitmaptile](#)
- [swf_shapefilloff](#)
- [swf_shapefillsolid](#)
- [swf_shapelinesolid](#)
- [swf_shapelineto](#)
- [swf_shapemoveto](#)
- [swf_showframe](#)
- [swf_startbutton](#)
- [swf_startdoaction](#)
- [swf_startshape](#)
- [swf_startsymbol](#)
- [swf_textwidth](#)
- [swf_translate](#)
- [swf_viewport](#)

8.128.9 swf_actiongotoframe() : Joue un cadre puis stoppe

void **swf_actiongotoframe** (int framenumbers)

[swf_actiongotoframe](#) se déplace jusqu'au cadre framenumbers , le joue, puis s'arrête.

8.128.10 swf_actiongotolabel() : Joue le cadre Flash spécifié

void **swf_actiongotolabel** (string label)

[swf_actiongotolabel](#) affiche le cadre appelé label , puis s'arrête.

8.128.11 swf_actionnextframe() : Avance d'un cadre

void **swf_actionnextframe** (void)

[swf_actionnextframe](#) avance d'un cadre le cadre courant.

8.128.12 swf_actionplay() : Joue l'animation Flash à partir du cadre courant

void **swf_actionplay** (void)

swf_actionplay joue l'animation Flash à partir du cadre courant.

8.128.13 swf_actionprevframe() : Recule d'un cadre

void **swf_actionprevframe** (void)

swf_actionnextframe recule d'un cadre le cadre courant.

8.128.14 swf_actionsettarget() : Fixe le contexte des actions

void **swf_actionsettarget** (string target)

swf_actionsettarget fixe le contexte des actions. Vous pouvez utiliser cette fonction pour contrôler d'autres animations Flash qui seraient en fonctionnement.

8.128.15 swf_actionstop() : Arrête l'animation Flash

void **swf_actionstop** (void)

swf_actionstop arrête l'animation Flash au cadre courant.

8.128.16 swf_actiontogglequality() : Choisit le niveau de qualité de l'animation Flash

void **swf_actiontogglequality** (void)

swf_actiontogglequality modifie le niveau de qualité de l'animation : haut ou bas.

8.128.17 swf_actionwaitforframe() : Ignore les actions si le cadre Flash n'est pas chargé

void **swf_actionwaitforframe** (int framenumbers , int skipcount)

swf_actionwaitforframe vérifie que le cadre framenumbers a bien été chargé. Si ce n'est pas le cas, elle ignore les actions skipcount . Cela est très utile pour les séquences du type "Chargement...".

8.128.18 swf_addbuttonrecord() : Contrôle la situation, l'apparence et la zone active du bouton Flash courant

void **swf_addbuttonrecord** (int states , int shapeid , int depth)

swf_addbuttonrecord permet de modifier les caractéristiques d'un bouton Flash. `states`, définit les états du bouton autorisés : ce peut être `BShitTest`, `BSDown`, `BSOver` ou `BSUp`. `shapeid` est l'apparence du bouton, c'est-à-dire l'objet qui représente le bouton. `depth` est la profondeur de placement du bouton, dans le cadre courant.

Exemple avec swf_addbuttonrecord

```
<?php
    swf_startButton ($objid, TYPE_MENUBUTTON);
    swf_addButtonRecord (BSDown|BSOver, $buttonImageId, 340);
    swf_onCondition (MenuEnter);
    swf_actionGetUrl ("http://www.designmultimedia.com", "_level1");
    swf_onCondition (MenuExit);
    swf_actionGetUrl ("", "_level1");
    swf_endButton ();
?>
```

8.128.19 swf_addcolor() : Fixe la couleur globale d'addition Flash (? : the global add color)

void **swf_addcolor** (float `r` , float `g` , float `b` , float `a`)

swf_mulcolor fixe la valeur globale de multiplication (the global multiply color...) à la couleur rgba . Cette couleur est utilisée (implicitement) par swf_placeobject , swf_modifyobject et swf_addbuttonrecord . La couleur d'un objet sera ajoutée à rgba lorsque l'objet est placé sur la scène.

Note

Les valeurs de rgba peuvent être positives ou négatives.

8.128.20 swf_closefile() : Ferme le fichier courant Shockwave Flash

void **swf_closefile** (int `return_file`)

swf_closefile ferme le fichier courant, qui a été ouvert avec swf_openfile . Si le paramètre `return_file` a été fourni, il contiendra le fichier SWF fermé.

Création d'un fichier Flash simple, basé sur une entrée de l'utilisateur, et sauvegarde dans une base.

```
<?php
// La variable $text est fournie par l'utilisateur
// Variables globales pour l'accès à la base de données
// utilisée dans la fonction wf_savedata()
$DBHOST = "localhost";
$DBUSER = "sterling";
$DBPASS = "secret";
swf_openfile ("php://stdout", 256, 256, 30, 1, 1, 1);
    swf_definefont (10, "Ligon-Bold");
        swf_fontsize (12);
        swf_fontslant (10);
    swf_definetext (11, $text, 1);
    swf_pushmatrix ();
        swf_translate (-50, 80, 0);
        swf_placeobject (11, 60);
    swf_popmatrix ();
    swf_showframe ();
    swf_startdoaction ();
        swf_actionstop ();
```

```

    swf_enddoaction ();
$data = swf_closefile (1);
$data ?
    swf_savedata ($data) :
    die ("Error could not save SWF file");
// void swf_savedata (string data)
// Sauve le fichier généré dans la base de données
// pour accès ultérieur
function swf_savedata ($data)
{
    global $DBHOST,
           $DBUSER,
           $DBPASS;
    $dbh = @mysql_connect ($DBHOST, $DBUSER, $DBPASS);
    if (!$dbh) {
        die (sprintf ("Error [%d]: %s",
                     mysql_errno (), mysql_error ()));
    }
    $stmt = "INSERT INTO swf_files (file) VALUES ('$data')";
    $sth = @mysql_query ($stmt, $dbh);
    if (!$sth) {
        die (sprintf ("Error [%d]: %s",
                     mysql_errno (), mysql_error ()));
    }
    @mysql_free_result ($sth);
    @mysql_close ($dbh);
}
?>

```

8.128.21 swf_definebitmap() : Définit une image bitmap

void **swf_definebitmap** (int objid , string image_name)

swf_definebitmap définit une bitmap à partir d'une image au format GIF, JPEG, RGB ou FI. L'image sera convertie en Flash JPEG ou Flash color map.

8.128.22 swf_definefont() : Définit une police Flash

void **swf_definefont** (int fontid , string fontname)

swf_definefont définit la police fontname et lui affecte l'identifiant fontid . Cette police devient alors la police courante.

8.128.23 swf_defineline() : Définit une ligne

void **swf_defineline** (int objid , float x1 , float y1 , float x2 , float y2 , float width)

swf_defineline définit une ligne commençant aux coordonnées (x1 , y1), et finissant au point de coordonnées (x2 , y2). Elle aura la largeur de width .

8.128.24 swf_definepoly() : Définit un polygone Flash

void **swf_definepoly** (int objid , array coords , int npoints , float width)

swf_definepoly définit un polygone, dont les coordonnées des sommets sont placées dans le tableau coords). npoints est le nombre de points contenus dans le tableau coords . width est la largeur des bords du polygone. Si width vaut 0.0, le polygone sera rempli.

8.128.25 swf_definerect() : Définit un rectangle

void **swf_definerect** (int objid , float x1 , float y1 , float x2 , float y2 , float width)

swf_definerect définit un rectangle, de coin supérieur gauche aux coordonnées (x1 , y1), et de coin inférieur droit aux coordonnées (x2 , y2). L'épaisseur des bords est donnée par le paramètre width . width , 0.0 le rectangle sera rempli.

8.128.26 swf_definetext() : Définit une chaîne de texte

void **swf_definetext** (int objid , string str , int docenter)

swf_definetext définit la chaîne de texte str , en utilisant la police courante. docenter indique si la chaîne doit être centrée (valeur de 1), ou pas.

8.128.27 swf_endbutton() : Termine la définition du bouton Flash courant

void **swf_endbutton** (void)

swf_endbutton termine la définition du bouton courant.

8.128.28 swf_enddoaction() : Termine l'action courante

void **swf_enddoaction** (void)

swf_startdoaction termine l'action courante, démarrée par swf_startdoaction .

8.128.29 swf_endshape() : Complète la définition de la forme Flash courante

void **swf_endshape** (void)

swf_endshape complète la définition de la forme courante.

8.128.30 swf_endsymbol() : Termine la définition de symbole

void **swf_endsymbol** (void)

swf_endsymbol termine la définition de symbole, qui a été commencée avec swf_startsymbol .

8.128.31 swf_fontsize() : Change la taille de la police

void **swf_fontsize** (float size)

swf_fontsize remplace la taille de la police par la taille size .

8.128.32 swf_fontslant() : Change l'inclinaison de la police courante

void **swf_fontslant** (float slant)

swf_fontslant fixe l'inclinaison de la police courante à slant . Les valeurs positives créeront une inclinaison vers la droite, les négatives, vers la gauche.

8.128.33 swf_fontracking() : Change l'espacement des caractères

void **swf_fontracking** (float tracking)

swf_fontracking change l'espacement, et lui affecte la valeur de tracking . Cette fonction sert à accroître l'espace entre les lettres et le texte. Les valeurs positives accroissent cet espace, les négatives le réduisent.

8.128.34 swf_getbitmapinfo() : Lit les informations sur une image

array **swf_getbitmapinfo** (int bitmapid)

swf_getbitmapinfo retourne un tableau d'informations sur l'image bitmap repérée par bitmapid . Le tableau contient les éléments suivants :

- " size " : la taille en octets de l'image.
- " width " : la largeur en pixels de l'image.
- " height " : la hauteur en pixels de l'image.

8.128.35 swf_getfontinfo() : Retourne la hauteur du A majuscule, et du x minuscule

array **swf_getfontinfo** (void)

swf_getfontinfo retourne la hauteur du A majuscule, et du x minuscule, dans un tableau associatif :

- " Aheight " : la hauteur du A majuscule, en pixels.
- " xheight " : la hauteur du x minuscule, en pixels.

8.128.36 swf_getframe() : Retourne le numéro de cadre courant

int **swf_getframe** (void)

swf_getframe retourne le numéro de cadre courant.

8.128.37 `swf_labelframe()` : Nomme le cadre courant

void `swf_labelframe` (string name)

`swf_labelframe` donne le nom name au cadre courant.

8.128.38 `swf_lookat()` : Définit une transformation de vue

void `swf_lookat` (float view_x , float view_y , float view_z , float reference_x , float reference_y , float reference_z , float twist)

`swf_lookat` définit une transformation de vue, en donnant la position de la vue, de coordonnées (view_x , view_y et view_z) et les coordonnées du point de référence dans la scène, de coordonnées (reference_x , reference_y , reference_z). Le paramètre twist contrôle la rotation le long de l'axe z de l'utilisateur.

8.128.39 `swf_modifyobject()` : Modifie un objet Flash

void `swf_modifyobject` (int depth , int how)

`swf_modifyobject` modifie la position et/ou la couleur de l'objet situé à la profondeur de depth . L'argument how détermine ce qui doit être modifié. how peut prendre les valeurs de MOD_MATRIX , MOD_COLOR ou la combinaison des deux.

MOD_COLOR utilise la couleur courante de multiplication (spécifiée par `swf_mulcolor`) et la couleur courante d'addition (spécifiée par `swf_addcolor`) pour colorer l'objet, et MOD_MATRIX utilise la matrice courante pour positionner l'objet.

8.128.40 `swf_mulcolor()` : Fixe la couleur globale de multiplication (? : the global multiply color)

void `swf_mulcolor` (float r , float g , float b , float a)

`swf_mulcolor` fixe la valeur globale de multiplication (the global multiply color...) à la couleur rgba . Cette couleur est utilisée (implicitement) par `swf_placeobject` , `swf_modifyobject` et `swf_addbuttonrecord` . La couleur d'un objet sera multipliée par rgba lorsque l'objet est placé sur la scène.

Note

Les valeurs de rgba peuvent être positives ou négatives.
--

8.128.41 `swf_nextid()` : Retourne le prochain identifiant d'objet libre

int `swf_nextid` (void)

`swf_nextid` retourne le prochain identifiant d'objet libre.

8.128.42 `swf_oncondition()` : Décrit la transition utilisée pour déclencher une liste d'actions

void `swf_oncondition` (int transition)

`swf_oncondition` décrit une transition qui va déclencher une liste d'actions. Il y a plusieurs types de transitions possibles, les suivantes sont destinées aux boutons de type `TYPE_MENUBUTTON` :

- `IdletoOverUp`
- `OverUptoldle`
- `OverUptoOverDown`
- `OverDowntoOverUp`
- `IdletoOverDown`
- `OutDowntoldle`
- `MenuEnter` (`IdletoOverUp|IdletoOverDown`)
- `MenuExit` (`OverUptoldle|OverDowntoldle`)

Pour les types `TYPE_PUSHBUTTON` voici les options :

- `IdletoOverUp`
- `OverUptoldle`
- `OverUptoOverDown`
- `OverDowntoOverUp`
- `OverDowntoOutDown`
- `OutDowntoOverDown`
- `OutDowntoldle`
- `ButtonEnter` (`IdletoOverUp|OutDowntoOverDown`)
- `ButtonExit` (`OverUptoldle|OverDowntoOutDown`)

8.128.43 `swf_openfile()` : Ouvre un nouveau fichier Shockwave Flash

void `swf_openfile` (string filename , float width , float height , float framerate , float r , float g , float b)

`swf_openfile` crée un nouveau fichier `filename` de largeur `width` , et de hauteur `height` , à la vitesse de `framerate` , de couleur de fond RGB (`r` , `g` , `b`).

`swf_openfile` doit être la première fonction à appeler, sous peine d'erreur mémoire (segmentation fault). Si vous voulez envoyer votre production au client HTML, utilisez le nom de fichier "php://stdout" (le support de ceci est prévue pour la version 4.0.1 et ultérieure).

8.128.44 `swf_ortho()` : Définit une projection orthogonale entre les coordonnées utilisateur et le port courant

void `swf_ortho` (float xmin , float xmax , float ymin , float ymax , float zmin , float zmax)

`swf_ortho` définit une projection orthogonale entre les coordonnées utilisateur et le port courant.

8.128.45 `swf_ortho2()` : Définit une projection orthogonale à 2 dimensions entre les coordonnées utilisateur et le port courant

void `swf_ortho2` (float `xmin` , float `xmax` , float `ymin` , float `ymax`)

`swf_ortho2` définit une projection orthogonale à 2 dimensions entre les coordonnées utilisateur et le port courant. C'est la projection par défaut des animations Flash. Si vous souhaitez une perspective, utilisez plutôt `swf_perspective` .

8.128.46 `swf_perspective()` : Définit une projection orthogonale à 3 dimensions entre les coordonnées utilisateur et le port courant

void `swf_perspective` (float `fovy` , float `aspect` , float `near` , float `far`)

`swf_perspective` définit une projection orthogonale à 3 dimensions entre les coordonnées utilisateur et le port courant. Le paramètre `fovy` est l'angle de vue de la direction y. Le paramètre `aspect` doit être choisi pour correspondre au ratio de la vue utilisée. `near` est le plan adjacent proche `far` est le plan adjacent distant.

Note

Diverses distorsions peuvent apparaître lors de ce genre de projections, car Flash ne dispose que d'une matrice à 2 dimensions. Certaines distorsions font vraiment tâche d'encre.
--

8.128.47 `swf_placeobject()` : Place un objet sur la scène

void `swf_placeobject` (int `objid` , int `depth`)

`swf_placeobject` place l'objet `objid` dans le cadre courant, à la profondeur `depth` . `objid` et `depth` doivent être compris entre 1 et 65535.

`swf_placeobject` utilise la couleur courante de multiplication (spécifiée par `swf_mulcolor`) et la couleur courante d'addition (spécifiée par `swf_addcolor`) pour colorer l'objet, et utilise la matrice courante pour positionner l'objet.

Note

Le support des couleurs RGBA est complet.

8.128.48 `swf_polarview()` : Définit le point de vue de l'utilisateur en coordonnées polaires

void `swf_polarview` (float `dist` , float `azimuth` , float `incidence` , float `twist`)

`swf_polarview` définit la position de l'utilisateur en coordonnées polaires. `dist` est la distance entre le point de vue et l'origine. `azimuth` définit l'angle azimutal dans le plan x,y mesuré en distance depuis l'axe y. `incidence` définit l'angle d'incidence dans le plan y,z, mesuré en distance depuis l'axe z. Finalement, `twist` est l'angle de rotation du point de vue sur la ligne de vue, en utilisant la règle de la main droite.

8.128.49 swf_popmatrix() : Dépile la matrice de transformation

void **swf_popmatrix** (void)

swf_popmatrix dépile la matrice de transformation.

8.128.50 swf_posround() : Active l'approximation des translations d'objets Flash

void **swf_posround** (int round)

swf_posround active ou désactive l'approximation lors des translations, lorsque des objets sont placés ou déplacés. Il y a des situations où le texte devient plus lisible lorsque l'approximation a été activée. round active l'approximation (1) ou la désactive (0).

8.128.51 swf_pushmatrix() : Empile la matrice de transformations courante dans la pile

void **swf_pushmatrix** (void)

swf_pushmatrix empile la matrice de transformations courante dans la pile.

8.128.52 swf_removeobject() : Enlève un objet

void **swf_removeobject** (int depth)

swf_removeobject enlève l'objet situé à la profondeur depth de la scène.

8.128.53 swf_rotate() : Rotation de la transformation courante

void **swf_rotate** (float angle , string axis)

swf_rotate fait subir la rotation d'angle angle , autour de l'axe axis . Les valeurs possibles pour axis sont : ' x ' (abscisses), ' y ' (ordonnées) ou ' z ' (cote).

8.128.54 swf_scale() : Homothétie

void **swf_scale** (float x , float y , float z)

swf_scale fait une mise à l'échelle de x pour les coordonnées x, de y pour les coordonnées y et z pour les coordonnées z.

8.128.55 swf_setfont() : Change la police courante

void **swf_setfont** (int fontid)

swf_setfont remplace la police courante par la police repérée par l'identifiant fontid .

8.128.56 swf_setframe() : Fixe le cadre courant

void **swf_setframe** (int framenummer)

swf_setframe sélectionne le cadre framenummer comme cadre actif.

8.128.57 swf_shapearc() : Dessine un arc de cercle

void **swf_shapearc** (float x , float y , float r , float ang1 , float ang2)

swf_shapearc dessine un arc de cercle, depuis l'angle ang1 jusqu'à l'angle ang2 . Le centre du cercle est aux coordonnées (x , y), et de rayon r .

8.128.58 swf_shapecurveto() : Dessine une courbe de Bézier quadratique entre deux points

void **swf_shapecurveto** (float x1 , float y1 , float x2 , float y2)

swf_shapecurveto dessine la courbe de Bézier quadratique entre les points de coordonnées (x1 , y1) et (x2 , y2). La position courante devient alors (x2 , y2).

8.128.59 swf_shapecurveto3() : Dessine une courbe Bézier cubique

void **swf_shapecurveto3** (float x1 , float y1 , float x2 , float y2 , float x3 , float y3)

Dessine une courbe de Bézier cubique, en utilisant les points de coordonnées (x1 , y1) et (x2 , y2) comme points de contrôle et le point de coordonnées (x3 , y3) comme point final. La position finale devient alors la position courante.

8.128.60 swf_shapefillbitmapclip() : Choisit le mode de remplissage Flash par texture

void **swf_shapefillbitmapclip** (int bitmapid)

Choisit le mode de remplissage par texture : les espaces vides seront remplis avec la bitmap bitmapid .

8.128.61 swf_shapefillbitmaptile() : Choisit le mode de remplissage par texture Flash répétée

void **swf_shapefillbitmaptile** (int bitmapid)

swf_shapefillbitmaptile active le mode de remplissage par texture : les espaces vides seront remplis avec la bitmap bitmapid , répétée autant de fois qu'il le faut (mode carrelage).

8.128.62 swf_shapefilloff() : Désactive le remplissage de formes Flash

void **swf_shapefilloff** (void)

swf_shapefilloff désactive le remplissage de la forme courante.

8.128.63 swf_shapefillsolid() : Fixe la couleur pour le style courant de remplissage Flash

void **swf_shapefillsolid** (float r , float g , float b , float a)

swf_shapefillsolid fixe la couleur pour le style courant de remplissage à rgba .

8.128.64 swf_shapelinesolid() : Fixe le style courant de ligne

void **swf_shapelinesolid** (float r , float g , float b , float a , float width)

swf_shapelinesolid permet de choisir le style de ligne, à savoir la couleur et la largeur. Si width vaut 0.0, les lignes ne seront pas dessinées.

8.128.65 swf_shapelineto() : Dessine une ligne

void **swf_shapelineto** (float x , float y)

swf_shapelineto dessine une ligne entre la position courante et le point de coordonnées (x , y). La position courante devient alors (x , y).

8.128.66 swf_shapemoveto() : Change la position courante

void **swf_shapemoveto** (float x , float y)

swf_shapemoveto fixe la position courante au point de coordonnées (x , y).

8.128.67 swf_showframe() : Affiche le cadre courant

void **swf_showframe** (void)

swf_showframe affiche le cadre courant.

8.128.68 swf_startbutton() : Commence la définition d'un bouton

void **swf_startbutton** (int objid , int type)

swf_startbutton commence la définition d'un bouton. type peut prendre les valeurs de TYPE_MENUBUTTON ou TYPE_PUSHBUTTON. La constante TYPE_MENUBUTTON permet au focus de traverser lorsque la souris est cliquée, alors que TYPE_PUSHBUTTON ne le permet pas.

8.128.69 swf_startdoaction() : Commence la description d'une liste d'actions pour le cadre courant

void **swf_startdoaction** (void)

swf_startdoaction commence la description d'une liste d'actions pour le cadre courant. Cette fonction doit être appelée avant que les actions ne soient définies pour le cadre courant.

8.128.70 swf_startshape() : Commence une forme complexe

void **swf_startshape** (int objid)

swf_startshape commence une forme complexe, qui sera repérée par l'identifiant d'objet objid .

8.128.71 swf_startsymbol() : Définit un symbole

void **swf_startsymbol** (int objid)

swf_startsymbol définit un identifiant d'objet comme symbole. Les symboles sont des petites animations Flash qui peuvent être jouées simultanément. objid est l'identifiant d'objet que vous voulez définir comme symbole.

8.128.72 swf_textwidth() : Retourne la longueur d'une chaîne

float **swf_textwidth** (string str)

swf_textwidth retourne la longueur de la chaîne str , en pixels, en utilisant la police courante.

8.128.73 swf_translate() : Translate la transformation courante

void **swf_translate** (float x , float y , float z)

swf_translate déplace la transformation courante de x , y et z , dans les directions x, y et z.

8.128.74 swf_viewport() : Sélectionne une nouvelle zone pour un dessin ultérieur

void **swf_viewport** (float xmin , float xmax , float ymin , float ymax)

swf_viewport sélectionne une nouvelle zone pour y dessiner ultérieurement. La zone est définie de xmin à xmax et de ymin à ymax . Si cette fonction n'est pas appelée, les valeurs par défaut sont celles de l'écran courant.

8.129 Sybase

8.129.1 Introduction

8.129.2 Pré-requis

8.129.3 Installation

Pour activer le support des bases de données Sybase-DB, vous devez compiler PHP avec l'option `--with-sybase[=DIR]` . DIR est le dossier d'installation de Sybase et, par défaut, il vaut `/home/sybase` . Pour activer le support de Sybase-CT, vous devez compiler PHP avec l'option `--with-sybase-ct[=DIR]` . DIR est le dossier d'installation de Sybase et, par défaut, il vaut `/home/sybase` .

8.129.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Sybase

Nom	Par défaut	Modifiable	Historique
<code>sybase.allow_persistent</code>	"1"	PHP_INI_ALL	Disponible depuis PHP 4.0.3.
<code>sybase.max_persistent</code>	"-1"	PHP_INI_ALL	Disponible depuis PHP 4.0.3.
<code>sybase.max_links</code>	"-1"	PHP_INI_ALL	Disponible depuis PHP 4.0.3.
<code>sybase.interface_file</code>	<code>"/usr/sybase/interfaces"</code>	PHP_INI_SYSTEM	
<code>sybase.min_error_severity</code>	"10"	PHP_INI_ALL	
<code>sybase.min_message_severity</code>	"10"	PHP_INI_ALL	
<code>sybase.compatability_mode</code>	"0"	PHP_INI_ALL	
<code>magic_quotes_sybase</code>	"0"	PHP_INI_ALL	

Voici un éclaircissement sur l'utilisation des directives de configuration.

`sybase.allow_persistent` booléen

Activation ou non des connexions persistantes.

`sybase.max_persistent` entier

Le nombre maximum de connexions Sybase persistantes par processus. -1 signifie illimité.

`sybase.max_links` entier

Le nombre maximum de connexions Sybase par processus. -1 signifie illimité.

`sybase.min_error_severity` entier

Le niveau minimal d'erreur à afficher.

`sybase.min_message_severity` entier

Le niveau minimal de message d'erreur à afficher.

`sybase.compatability_mode` booléen

Mode de compatibilité avec les anciennes versions de PHP 3.0. Si activé (on), cela va faire que PHP assigne automatiquement des types aux résultats, en fonctions de leur type Sybase, au lieu de les traiter comme des chaînes de caractères. Ce mode de compatibilité ne va pas rester des années, alors essayez de modifier votre code, et de désactiver cette fonctionnalité.

`magic_quotes_sybase` booléen

Si `magic_quotes_sybase` est aussi activé, un guillemets simple est échappé avec un guillemet simple au lieu d'un anti-slash, si `magic_quotes_gpc` ou `magic_quotes_runtime` est activé.

Note
Si la directive <code>magic_quotes_sybase</code> est activée, elle remplacera complètement <code>magic_quotes_gpc</code> . Ce qui fait que même si <code>magic_quotes_gpc</code> retourne TRUE , les guillemets doubles, les anti-slashes ou les caractères NULL ne seront pas protégés.

Sybase-CT

Nom	Par défaut	Modifiable	Historique
<code>sybct.allow_persistent</code>	"1"	PHP_INI_SYSTEM	Disponible depuis PHP 4.0.4.
<code>sybct.max_persistent</code>	"-1"	PHP_INI_SYSTEM	Disponible depuis PHP 4.0.4.
<code>sybct.max_links</code>	"-1"	PHP_INI_SYSTEM	Disponible depuis PHP 4.0.4.
<code>sybct.min_server_severity</code>	"10"	PHP_INI_ALL	Disponible depuis PHP 4.0.4.
<code>sybct.min_client_severity</code>	"10"	PHP_INI_ALL	Disponible depuis PHP 4.0.4.
<code>sybct.hostname</code>	NULL	PHP_INI_ALL	Disponible depuis PHP 4.0.4.
<code>sybct.deadlock_retry_count</code>	"0"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.

Voici un éclaircissement sur l'utilisation des directives de configuration.

`sybct.allow_persistent` booléen

Autorise ou pas les connexions persistantes Sybase-CT. Par défaut vaut On .

`sybct.max_persistent` entier

Le nombre maximum de connexions Sybase-CT persistantes par processus. -1 signifie illimité. C'est aussi la valeur par défaut.

`sybct.max_links` entier

Le nombre maximum de connexions Sybase-CT par processus, y compris les connexions persistantes. -1 signifie illimité.

`sybct.min_server_severity` entier

Les messages du serveur ayant une criticité supérieure ou égale à la valeur de cette option seront rapportés comme alertes. Cette valeur peut aussi être modifiée depuis les scripts en appelant la fonction `sybase_min_server_severity` . Par défaut, cette option vaut 10.

`sybct.min_client_severity` entier

Les messages de la bibliothèque cliente ayant une criticité supérieure ou égale à la valeur de cette option seront rapportés comme alertes. Cette valeur peut aussi être modifiée depuis les scripts en appelant la fonction `sybase_min_client_severity` . Par défaut, cette option vaut 10.

`sybct.login_timeout` entier

Le nombre maximal de secondes à attendre pour effectuer avec succès une connexion avant d'échouer. Notez que si `max_execution_time` est dépassé avant que ce nombre de secondes, votre script se terminera avant de pouvoir exécuter une action sur l'échec de connexion. Par défaut, ce nombre vaut une minute.

`sybct.timeout` entier

Le nombre maximal de secondes à attendre pour effectuer avec succès une requête avant d'échouer. Notez que si `max_execution_time` est dépassé avant que ce nombre de secondes, votre script se terminera avant de pouvoir exécuter une action sur l'échec de connexion. Par défaut, aucune limite n'est fixé.

`sybct.hostname` chaîne de caractères

Le nom de l'hôte à partir duquel vous prétendez être connecté, afin qu'il soit affiché par `sp_who`. Par défaut, il n'y a pas de valeur.

`sybct.deadlock_retry_count` int

Permet de définir le nombre de tentatives de résolution des blocages. La valeur par défaut est -1, ou `forever`, c'est à dire indéfiniment.

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#).

8.129.5 Types de ressources

8.129.6 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [sybase_affected_rows](#)
- [sybase_close](#)
- [sybase_connect](#)
- [sybase_data_seek](#)
- [sybase_deadlock_retry_count](#)
- [sybase_fetch_array](#)
- [sybase_fetch_assoc](#)
- [sybase_fetch_field](#)
- [sybase_fetch_object](#)
- [sybase_fetch_row](#)
- [sybase_field_seek](#)
- [sybase_free_result](#)
- [sybase_get_last_message](#)
- [sybase_min_client_severity](#)
- [sybase_min_error_severity](#)
- [sybase_min_message_severity](#)
- [sybase_min_server_severity](#)
- [sybase_num_fields](#)
- [sybase_num_rows](#)
- [sybase_pconnect](#)
- [sybase_query](#)
- [sybase_result](#)
- [sybase_select_db](#)
- [sybase_set_message_handler](#)
- [sybase_unbuffered_query](#)

8.129.8 `sybase_close()` : Ferme une connexion Sybase

bool `sybase_close` (*resource link_identifieur*)

`sybase_close` termine la connexion avec le serveur Sybase associé à l'identifiant de connexion `link_identifieur` .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Notez qu'il n'est pas utile de fermer les connexions non persistantes, car elles seront terminées à la fin du script.

`sybase_close` ne ferme pas les connexions persistantes générées par `sybase_pconnect` .

Voir aussi `sybase_connect` et `sybase_pconnect` .

8.129.9 `sybase_connect()` : Ouvre une connexion à un serveur Sybase

resource `sybase_connect` (*string servername* , *string username* , *string password* , *string charset* , *string appname*)

`sybase_connect` retourne un identifiant positif de lien Sybase en cas de succès, et FALSE en cas d'erreur.

`sybase_connect` établit une connexion à un serveur Sybase. Le nom de serveur `servername` doit être valide, défini dans le fichier d'interface.

Si un deuxième appel à `sybase_connect` est fait avec les mêmes arguments, une nouvelle connexion ne sera pas établie, mais ce sera l'identifiant de la connexion déjà ouverte qui sera retourné.

La connexion sera fermée dès la fin du script, à moins qu'elle ne soit pas explicitement fermée avec `sybase_close` .

Exemple avec `sybase_connect`

```
<?php
$link = sybase_connect('SYBASE', '', '')
or die("Impossible de se connecter !");
print ("Connexion établie");
sybase_close($link);
?>
```

Voir aussi `sybase_pconnect` et `sybase_close` .

8.129.10 `sybase_data_seek()` : Déplace le pointeur interne de lignes Sybase

bool `sybase_data_seek` (*resource result_identifieur* , *int row_number*)

`sybase_data_seek` déplace le pointeur interne de lignes du résultat Sybase associé à `result_identifieur` jusqu'à la ligne `row_number` . Le prochain appel à `sybase_fetch_row` sans préciser la ligne, retournera la ligne `row_number` .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [sybase_fetch_row](#) .

8.129.11 sybase_deadlock_retry_count() : Configure le nombre de tentatives lors de blocages (deadlock)

void **sybase_deadlock_retry_count** (int retry_count)

[sybase_deadlock_retry_count](#) permet de configurer à `retry_count` le nombre d'essais que Sybase va tenter lorsqu'il rencontre un blocage de verrou (deadlocks). Par défaut, tous les blocages sont testés un nombre infini de fois, ou bien jusqu'à ce que le script soit interrompu (par exemple, par la fonction [set_time_limit](#)) ou que la requête réussisse.

Note
Cette fonction n'est disponible qu'avec la bibliothèque CT Sybase, et non pas avec la bibliothèque DB.

-1	Essai jusqu'à réussir (par défaut)
0	Ne pas réessayer
n	Réessaye n fois

8.129.12 sybase_fetch_array() : Retourne une ligne Sybase sous la forme d'un tableau

array **sybase_fetch_array** (resource result)

[sybase_fetch_array](#) retourne un tableau qui contient la ligne demandée, ou FALSE s'il ne reste plus de ligne.

[sybase_fetch_array](#) est une version évoluée de [sybase_fetch_row](#) . En plus d'enregistrer les données dans un tableau à index numérique, cette fonction peut aussi les enregistrer dans un tableau associatif, en utilisant les noms des champs comme clés.

Il est très important de noter que [sybase_fetch_array](#) N'est PAS nettement plus lente que [sybase_fetch_row](#) , tandis qu'elle fournit un confort d'utilisation notable.

Note
Lors de sélection de champs avec des noms identiques (par exemple, dans un join), les index associatifs auront un nombre séquentiel ajouté au début. Voir l'exemple pour plus de détails.
Cas des colonnes de noms identiques avec Sybase
<pre> <?php \$dbh= sybase_connect('SYBASE', '', ''); \$q= sybase_query('SELECT * FROM p, a WHERE p.person_id= a.person_id', \$dbh); var_dump(sybase_fetch_array(\$q)); sybase_close(\$dbh); ?> </pre>

L'exemple ci-dessus va produire l'affichage suivant (en supposant que les tables ont toutes les deux des colonnes appelées `person_id`) :

```
array(4) {
  [0]=>
  int(1)
  ["person_id"]=>
  int(1)
  [1]=>
  int(1)
  ["person_id1"]=>
  int(1)
}
```

Voir aussi [sybase_fetch_row](#) , [sybase_fetch_assoc](#) et [sybase_fetch_object](#) .

8.129.13 `sybase_fetch_assoc()` : Lit une ligne de résultat Sybase sous forme de tableau associatif

array `sybase_fetch_assoc` (resource result)

`sybase_fetch_assoc` retourne un tableau contenant la ligne lue, ou bien FALSE s'il n'y a plus de lignes.

Note

Cette fonction n'est disponible qu'avec la bibliothèque CT Sybase, et non pas avec la bibliothèque DB.

`sybase_fetch_assoc` est une alternative de `sybase_fetch_row` qui utilise les noms de colonnes plutôt que des entiers comme index dans le tableau de résultat. Les colonnes de différentes tables ayant le même nom sont retournées sous la forme `name`, `name1`, `name2`, ..., `nameN`.

Il est important de noter qu'utiliser `sybase_fetch_assoc` n'est pas significativement plus lent que `sybase_fetch_row` , tout en procurant un niveau de confort accru.

Voir aussi [sybase_fetch_array](#) , [sybase_fetch_object](#) et [sybase_fetch_row](#) .

8.129.14 `sybase_fetch_field()` : Lit les informations d'un champ Sybase

object `sybase_fetch_field` (resource result , int field_offset)

`sybase_fetch_field` retourne un objet contenant les informations du champ.

`sybase_fetch_field` sert à obtenir des informations à propos des champs dans le résultat `result` . Si l'offset du champ n'est pas précisé, le champ suivant est traité.

Les propriétés des objets sont :

- `name` - column name. nom de la colonne. Si la colonne est un résultat de fonction, le nom de cette fonction devient `computed#N`, où #N est un numéro de série.
- `column_source` - la table d'origine de la colonne.
- `max_length` - taille maximale de la colonne

- numeric - 1 si la colonne est de type numérique.
- type - type de données de la colonne

Voir aussi [sybase_field_seek](#) .

8.129.15 `sybase_fetch_object()` : Retourne une ligne Sybase sous la forme d'un objet

objet `sybase_fetch_object` (resource result , *mixed* object)

`sybase_fetch_object` retourne un objet qui contient la ligne demandée, en cas de succès, et FALSE en cas d'erreur.

`sybase_fetch_object` est similaire à `sybase_fetch_array` , avec une différence : c'est un objet qui est retourné à la place d'un tableau. Indirectement, cela signifie que vous ne pourrez accéder aux valeurs que par les propriétés, et non plus avec des offsets (les nombres sont interdits comme nom de propriété).

Utilisez le second paramètre `object` pour spécifier le type d'objet que vous voulez retourner. Si ce paramètre est omis, l'objet résultant sera de classe `stdClass` .

<p>Note</p> <p>Depuis PHP 4.3.0, cette fonction ne retourne plus les membres numériques.</p> <p>Ancienne méthode :</p> <pre>object(stdclass)(3) { [0]=> string(3) "foo" ["foo"]=> string(3) "foo" [1]=> string(3) "bar" ["bar"]=> string(3) "bar" }</pre> <p>Nouvelle méthode :</p> <pre>object(stdclass)(3) { ["foo"]=> string(3) "foo" ["bar"]=> string(3) "bar" }</pre>
<p><code>sybase_fetch_object</code> retourne l'objet Foo</p> <pre><?php class Foo { var \$foo, \$bar, \$baz; } // {...} \$qrh= sybase_query('SELECT foo, bar, baz FROM example'); \$foo= sybase_fetch_object(\$qrh, 'Foo'); \$bar= sybase_fetch_object(\$qrh, new Foo()); // {...} ?></pre>

Au niveau de la vitesse, cette fonction est identique à [sybase_fetch_array](#) , et presque aussi rapide que [sybase_fetch_row](#) (la différence est insignifiante).

Voir aussi [sybase_fetch_array](#) et [sybase_fetch_row](#) .

8.129.16 [sybase_fetch_row\(\)](#) : Retourne une ligne Sybase sous la forme d'un tableau numérique

array [sybase_fetch_row](#) (resource result)

[sybase_fetch_row](#) retourne un tableau qui contient la ligne demandée, en cas de succès, et FALSE en cas d'erreur.

[sybase_fetch_row](#) lit une ligne dans le résultat associé à l'identifiant de résultat result . La ligne retournée est sous la forme d'un tableau. Chaque champ est enregistré dans un index du tableau, les index commençant à 0.

Les prochains appels à [sybase_fetch_row](#) retourneront la ligne suivante du résultat, ou FALSE , s'il ne reste plus de ligne.

PHP	Sybase
string	VARCHAR, TEXT, CHAR, IMAGE, BINARY, VARBINARY, DATETIME
int	NUMERIC (sans précision), DECIMAL (sans précision), INT, BIT, TINYINT, SMALLINT
float	NUMERIC (avec précision), DECIMAL (avec précision), REAL, FLOAT, MONEY
NULL	NULL

Voir aussi [sybase_result](#) , [sybase_fetch_array](#) , [sybase_fetch_assoc](#) , [sybase_fetch_object](#) et [sybase_data_seek](#) , [sybase_fetch_assoc](#) .

8.129.17 [sybase_field_seek\(\)](#) : Modifie l'index d'un champ Sybase

bool [sybase_field_seek](#) (resource result , int field_offset)

[sybase_field_seek](#) modifie l'index d'un champ. Le prochain appel à la fonction [sybase_fetch_field](#) sans préciser l'index du champ retournera ce champ.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi [sybase_fetch_field](#) .

8.129.18 [sybase_free_result\(\)](#) : Libère un résultat Sybase de la mémoire

bool [sybase_free_result](#) (resource result)

[sybase_free_result](#) n'est vraiment utile que si vous risquez d'utiliser trop de mémoire durant votre script. La mémoire occupée par les résultats est automatiquement libérée à la fin du script. Vous devez utiliser [sybase_free_result](#) avec un identifiant de résultat et la mémoire utilisée par celui-ci sera libérée.

8.129.19 `sybase_get_last_message()` : Retourne le dernier message du serveur

string `sybase_get_last_message` (void)

`sybase_get_last_message` retourne le dernier message rapporté par le serveur.

Voir aussi `sybase_min_message_severity` .

8.129.20 `sybase_min_client_severity()` : Fixe la sévérité minimale du client Sybase

void `sybase_min_client_severity` (int severity)

`sybase_min_client_severity` fixe la sévérité minimale du client.

Note
Cette fonction n'est disponible qu'avec la bibliothèque CT Sybase, et non pas avec la bibliothèque DB.

Voir aussi `sybase_min_server_severity` .

8.129.21 `sybase_min_error_severity()` : Fixe la sévérité minimale du client pour les erreurs

void `sybase_min_error_severity` (int severity)

`sybase_min_error_severity` fixe la sévérité minimale du client pour les erreurs.

Note
Cette fonction n'est disponible qu'avec la bibliothèque DB Sybase, et non pas avec la bibliothèque CT.

Voir aussi `sybase_min_message_severity` .

8.129.22 `sybase_min_message_severity()` : Fixe la sévérité minimale du client pour les messages

void `sybase_min_message_severity` (int severity)

`sybase_min_message_severity` fixe la sévérité minimale du client pour les messages.

Note
Cette fonction n'est disponible qu'avec la bibliothèque DB Sybase, et non pas avec la bibliothèque CT.

Voir aussi `sybase_min_error_severity` .

8.129.23 `sybase_min_server_severity()` : Fixe la sévérité minimale du client pour le serveur Sybase

void **sybase_min_server_severity** (int severity)

`sybase_min_server_severity` fixe la sévérité minimale du client pour le serveur.

Note
Cette fonction n'est disponible qu'avec la bibliothèque CT Sybase, et non pas avec la bibliothèque DB.

Voir aussi `sybase_min_client_severity` .

8.129.24 `sybase_num_fields()` : Retourne le nombre de champs dans un résultat Sybase

int **sybase_num_fields** (resource result)

`sybase_num_fields` retourne le nombre de champs du résultat result .

Voir aussi `sybase_db_query` , `sybase_query` , `sybase_fetch_field` et `sybase_num_rows` .

8.129.25 `sybase_num_rows()` : Retourne le nombre de lignes dans un résultat Sybase

int **sybase_num_rows** (resource result)

`sybase_num_rows` retourne le nombre de lignes du résultat result .

Voir aussi `sybase_num_fields` , `sybase_query` et `sybase_fetch_row` .

8.129.26 `sybase_pconnect()` : Ouvre une connexion persistante à un serveur Sybase

resource **sybase_pconnect** (*string* servername , *string* username , *string* password , *string* charset , *string* appname)

`sybase_pconnect` retourne un identifiant de connexion positif en cas de succès, et FALSE en cas d'erreur.

`sybase_connect` se comporte comme `sybase_pconnect` avec deux différences majeures :

Premièrement, lors de la connexion, la fonction va chercher une connexion (persistante) déjà ouverte, avec le même hôte, nom de compte et mot de passe. Si une telle connexion est trouvée, un identifiant de cette connexion est retourné, plutôt que d'en ouvrir une nouvelle.

Deuxièmement, la connexion au serveur Sybase ne sera pas terminée lors de la fin du script. Au contraire, le lien sera maintenu pour des connexions ultérieures. `sybase_close` ne fermera pas un lien créé par `sybase_pconnect` .

Ce type de lien est dit 'persistant'.

Voir aussi [sybase_connect](#) .

8.129.27 [sybase_query\(\)](#) : Envoie une requête à une base Sybase

mixed [sybase_query](#) (string query , resource link_identifier)

[sybase_query](#) retourne un identifiant de résultat positif en cas de succès, FALSE si une erreur survient, ou TRUE si la requête a été effectuée avec succès mais n'a retourné aucune colonne.

[sybase_query](#) envoie une requête à la base de données courante, sur le serveur associé à l'identifiant de connexion. Si l'identifiant de connexion n'est pas précisé, la fonction essaiera d'utiliser la dernière connexion ouverte. Si aucune connexion n'a été ouverte, la fonction va tenter d'ouvrir une connexion avec la fonction [sybase_connect](#) .

Voir aussi [sybase_select_db](#) et [sybase_connect](#) .

8.129.28 [sybase_result\(\)](#) : Lit une valeur dans un résultat

string [sybase_result](#) (resource result , int row , mixed field)

[sybase_result](#) retourne le contenu d'une cellule. L'argument field peut être l'index du champ, ou bien le nom du champ, ou encore le nom de la table " point " le nom du champ. Si la colonne a été aliasée ('SELECT foo AS bar FROM...'), utilisez l'alias à la place du nom de la colonne.

Lorsque vous travaillez sur des résultats de grande taille, vous devriez utiliser les autres fonctions qui lisent une ligne entière (voir plus loin). Etant donné que ces fonctions lisent une ligne entière, elles sont BEAUCOUP plus rapide que [sybase_result](#) . De plus, l'utilisation d'index numérique est beaucoup plus rapide que les noms des champs, ou les noms des tables et des champs.

Fonctions de substitution, à haute efficacité : [sybase_fetch_row](#) , [sybase_fetch_array](#) et [sybase_fetch_object](#) .

8.129.29 [sybase_select_db\(\)](#) : Sélectionne une base de données Sybase

bool [sybase_select_db](#) (string database_name , resource link_identifier)

[sybase_select_db](#) change la base de données courante et active sur le serveur associé avec l'identifiant de connexion link_identifier . Si link_identifier n'est pas précisé, le dernier lien ouvert est utilisé. Si aucun lien n'a été ouvert, la fonction va tenter d'en établir un en appelant [sybase_connect](#) .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Tous les prochains appels à [sybase_query](#) seront faits dans la base de données courante et active.

Voir aussi [sybase_connect](#) , [sybase_pconnect](#) et [sybase_query](#) .

8.129.30 `sybase_set_message_handler()` : Configure le gestionnaire de messages Sybase

bool `sybase_set_message_handler` (callback handler , *resource* connection)

`sybase_set_message_handler` configure la fonction handler pour qu'il soit le gestionnaire des messages générés par le serveur Sybase. Vous pouvez spécifier le nom d'une fonction globale, ou bien utiliser un tableau pour spécifier un objet et un nom de méthode.

Note

Cette fonction n'est disponible qu'avec la bibliothèque CT Sybase, et non pas avec la bibliothèque DB.

Le gestionnaire attend 5 arguments, dans l'ordre suivant : numéro de message, sévérité, état, numéro de ligne et description. Les 4 premiers paramètres sont des entiers et le dernier est une chaîne de caractères. Si ce gestionnaire retourne la valeur FALSE , PHP retournera un message d'erreur classique.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Le paramètre `connection` a été ajouté dans PHP 4.3.5.

Fonction de callback pour `sybase_set_message_handler`

```
<?php
function msg_handler($msgnumber, $severity, $state, $line, $text)
{
    var_dump($msgnumber, $severity, $state, $line, $text);
}

sybase_set_message_handler('msg_handler');
?>
```

Classe de callback pour `sybase_set_message_handler`

```
<?php
class Sybase {
    function handler($msgnumber, $severity, $state, $line, $text)
    {
        var_dump($msgnumber, $severity, $state, $line, $text);
    }
}

$sybase = new Sybase();
sybase_set_message_handler(array($sybase, 'handler'));
?>
```

Gestion des messages non traités de `sybase_set_message_handler`

```
<?php
// Retournez FALSE si vous voulez indiquer que vous ne savez pas
// traiter ce message. Cette erreur est alors affichée sous la forme
// d'une alerte, comme s'il n'y avait pas de gestionnaire installé.
function msg_handler($msgnumber, $severity, $state, $line, $text)
{
    if (257 == $msgnumber) {
        return FALSE;
    }
    var_dump($msgnumber, $severity, $state, $line, $text);
}

sybase_set_message_handler('msg_handler');
```

?>

8.129.31 `sybase_unbuffered_query()` : Envoie une requête à Sybase et ne bloque pas

resource **sybase_unbuffered_query** (string query , resource link_identifieur , bool store_result)

`sybase_unbuffered_query` retourne une ressource Sybase en cas de réussite, et FALSE sinon.

Note

Cette fonction n'est disponible qu'avec la bibliothèque CT Sybase, et non pas avec la bibliothèque DB.

`sybase_unbuffered_query` envoie la requête query au serveur Sybase actif, représenté par link_identifieur . Si link_identifieur n'est pas spécifié, la dernière connexion ouverte sera utilisée. Si aucune connexion n'a été ouverte, la fonction essaiera d'établir un lien avec `sybase_connect` , et de l'utiliser.

Contrairement à `sybase_query` , `sybase_unbuffered_query` lit uniquement la première ligne lue. `sybase_fetch_array` et les fonctions similaires lisent les lignes nécessaires. `sybase_data_seek` lit jusqu'à la prochaine ligne. Ce comportement fournit des performances accrues pour les résultats de grande taille.

`sybase_num_rows` ne retournera le nombre de lignes correct que si toutes les lignes du résultat ont été lues. Pour Sybase, le nombre de ligne n'est pas connu, et doit être calculé par le programme client.

Note

Si vous ne lisez pas toutes les lignes du résultat avant d'exécuter une nouvelle requête, PHP va générer une alerte, et annuler toutes les lignes qui restent à lire. Pour éviter cela, utilisez la fonction `sybase_free_result` qui va annuler les lignes restantes.

Le paramètre optionnel store_result peut valoir FALSE pour indiquer que les lignes ne doivent pas être ramenées en mémoire, afin de réduire la consommation, surtout dans le cas de très grands résultats.

Exemple avec `sybase_unbuffered_query`

```
<?php
$dbh= sybase_connect('SYBASE', '', '');
$q= sybase_unbuffered_query('select firstname, lastname from huge_table', $dbh, FALSE);
sybase_data_seek($q, 10000);
$i= 0;
while ($row= sybase_fetch_row($q)) {
    echo $row[0].' ' .$row[1];
    if ($i++ > 40000) break;
}
sybase_free_result($q);
sybase_close($dbh);
?>
```

Voir aussi `sybase_query` .

8.130 Fonctions TCP Wrappers

8.130.1 Introduction

L'extension TCP wrappers fournit un mécanisme classique Unix conçu pour vérifier si le client est capable de se connecter à partir d'une IP donnée.

8.130.2 Installation

Tcpwrap est actuellement disponible dans PECL <http://pecl.php.net/package/tcpwrap> .

Si PEAR est disponible sur votre système *nix vous pouvez utiliser l'installateur PEAR pour installer l'extension tcpwrap, avec la commande suivante : `pear -v install tcpwrap` .

Vous pouvez toujours télécharger le tar.gz et installer tcpwrap à la main :

Installation à la main de tcpwrap

```
gunzip tcpwrap-xxx.tgz
tar -xvf tcpwrap-xxx.tar
cd tcpwrap-xxx
phpize
./configure && make && make install
```

- [Introduction](#)
- [Installation](#)
- [tcpwrap_check](#)

8.131 Tidy

8.131.1 Introduction

Tidy est une interface avec la bibliothèque Tidy HTML, pour nettoyer et manipuler les documents HTML, et les traiter sous forme de balises hiérarchisées.

8.131.2 Pré-requis

Pour utiliser Tidy, vous devez disposer de la bibliothèque libtidy, qui est téléchargeable sur le site de <http://tidy.sourceforge.net/> .

8.131.3 Installation

Tidy est actuellement disponible pour PHP 4.3.x et PHP 5 en tant qu'extension PECL. L'extension est disponible sur <http://pecl.php.net/package/tidy> .

Note

Tidy 1.0 fonctionne juste avec PHP 4.3.x, alors que Tidy 2.0 ne fonctionne que avec PHP 5.

Si **PEAR** est disponible sur votre système *nix vous pouvez utiliser l'installateur PEAR pour avoir l'extension tidy, avec la commande suivante : `pear -v install tidy` .

Vous pouvez aussi télécharger l'archive tar.gz et installer tidy à la main :

Installation de tidy à la main pour PHP 4.3.x

```
gunzip tidy-xxx.tgz
tar -xvf tidy-xxx.tar
cd tidy-xxx
phpize
./configure && make && make install
```

Les utilisateurs de windows peuvent télécharger la dll de l'extension `php_tidy.dll` à partir de http://snaps.php.net/win32/PECL_STABLE/ .

En PHP 5 vous avez uniquement besoin de compiler en utilisant l'option `--with-tidy` .

8.131.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini` .

Nom	Par défaut	Modifiable	Historique
<code>tidy.default_config</code>	""	<code>PHP_INI_SYSTEM</code>	Disponible depuis PHP 5.0.0.
<code>tidy.clean_output</code>	"0"	<code>PHP_INI_PERDIR</code>	Disponible depuis PHP 5.0.0.

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

```
tidy.default_config string
    Chemin par défaut pour le fichier de configuration Tidy.
```

tidy.clean_output boolean

Activer ou désactiver la réparation du HTML par Tidy

Attention

N'activez pas tidy.clean_output si vous générez autre chose que du HTML, comme des images dynamiques.

8.131.5 Types de ressources

Cette extension ne définit aucune ressource.

8.131.6 Classes pré-définies

8.131.6.1 tidyNode

8.131.6.1.1 Méthodes

- [function.tidyNode-hasChildren](#) - Retourne TRUE si le noeud courant a des enfants
- [function.tidyNode-hasSiblings](#) - Retourne TRUE si le noeud courant a des frères
- [function.tidyNode-isAsp](#) - Retourne TRUE si le noeud courant est du ASP
- [function.tidyNode-isComment](#) - Retourne TRUE si le noeud courant est un commentaire
- [function.tidyNode-isHtml](#) - Retourne TRUE si le noeud courant est du code de l'HTML
- [function.tidyNode-isJste](#) - Retourne TRUE si le noeud courant est du JSTE
- [function.tidyNode-isPhp](#) - Retourne TRUE si le noeud courant est du PHP
- [function.tidyNode-isText](#) - Retourne TRUE si le noeud courant est du texte (aucun marquage)

8.131.6.1.2 Propriétés

- value - la valeur du noeud (par exemple, le texte html)
- name - le nom de la balise (par exemple, html, a, etc.)
- type - le type du noeud (une des constantes au-dessus, par exemple, TIDY_NODETYPE_PHP)
- line* - la ligne où le noeud commence
- column* - la colonne où le noeud commence
- proprietary* - TRUE si le noeud réfère à une balise propriétaire
- id - le ID de la balise (une des constantes au-dessus, par exemple, TIDY_TAG_FRAME)
- attribute - un tableau avec les attributs du noeud courant ou NULL s'il n'y en a pas
- child - un tableau avec les enfants tidyNode ou NULL s'il n'y en a pas

Note

Les propriétés marquées d'un * sont seulement disponibles à partir de PHP 5.1.0.

8.131.7 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Chaque TIDY_TAG_XXX représente un tag HTML. Par exemple, TIDY_TAG_A représente le tag link. Chaque TIDY_ATTR_XXX représente un attribut HTML. Par exemple, TIDY_ATTR_HREF représentera l'attribut href dans l'exemple précédent.

Les constantes suivantes sont définies par cette extension :

constante
TIDY_TAG_UNKNOWN
TIDY_TAG_A
TIDY_TAG_ABBR
TIDY_TAG_ACRONYM
TIDY_TAG_ALIGN
TIDY_TAG_APPLET
TIDY_TAG_AREA
TIDY_TAG_B
TIDY_TAG_BASE
TIDY_TAG_BASEFONT
TIDY_TAG_BDO
TIDY_TAG_BGSOUND
TIDY_TAG_BIG
TIDY_TAG_BLINK
TIDY_TAG_BLOCKQUOTE
TIDY_TAG_BODY
TIDY_TAG_BR
TIDY_TAG_BUTTON
TIDY_TAG_CAPTION
TIDY_TAG_CENTER
TIDY_TAG_CITE
TIDY_TAG_CODE
TIDY_TAG_COL
TIDY_TAG_COLGROUP
TIDY_TAG_COMMENT
TIDY_TAG_DD
TIDY_TAG_DEL
TIDY_TAG_DFN
TIDY_TAG_DIR
TIDY_TAG_DIV
TIDY_TAG_DL
TIDY_TAG_DT
TIDY_TAG_EM
TIDY_TAG_EMBED
TIDY_TAG_FIELDSET
TIDY_TAG_FONT
TIDY_TAG_FORM
TIDY_TAG_FRAME
TIDY_TAG_FRAMESET

TIDY_TAG_H1
TIDY_TAG_H2
TIDY_TAG_H3
TIDY_TAG_H4
TIDY_TAG_H5
TIDY_TAG_H6
TIDY_TAG_HEAD
TIDY_TAG_HR
TIDY_TAG_HTML
TIDY_TAG_I
TIDY_TAG_IFRAME
TIDY_TAG_ILAYER
TIDY_TAG_IMG
TIDY_TAG_INPUT
TIDY_TAG_INS
TIDY_TAG_ISINDEX
TIDY_TAG_KBD
TIDY_TAG_KEYGEN
TIDY_TAG_LABEL
TIDY_TAG_LAYER
TIDY_TAG_LEGEND
TIDY_TAG_LI
TIDY_TAG_LINK
TIDY_TAG_LISTING
TIDY_TAG_MAP
TIDY_TAG_MARQUEE
TIDY_TAG_MENU
TIDY_TAG_META
TIDY_TAG_MULTICOL
TIDY_TAG_NOBR
TIDY_TAG_NOEMBED
TIDY_TAG_NOFRAMES
TIDY_TAG_NOLAYER
TIDY_TAG_NOSAFE
TIDY_TAG_NOSCRIPT
TIDY_TAG_OBJECT
TIDY_TAG_OL
TIDY_TAG_OPTGROUP
TIDY_TAG_OPTION
TIDY_TAG_P
TIDY_TAG_PARAM
TIDY_TAG_PLAINTEXT
TIDY_TAG_PRE
TIDY_TAG_Q
TIDY_TAG_RP

TIDY_TAG_RT
TIDY_TAG_RTC
TIDY_TAG_RUBY
TIDY_TAG_S
TIDY_TAG_SAMP
TIDY_TAG_SCRIPT
TIDY_TAG_SELECT
TIDY_TAG_SERVER
TIDY_TAG_SERVLET
TIDY_TAG_SMALL
TIDY_TAG_SPACER
TIDY_TAG_SPAN
TIDY_TAG_STRIKE
TIDY_TAG_STRONG
TIDY_TAG_STYLE
TIDY_TAG_SUB
TIDY_TAG_TABLE
TIDY_TAG_TBODY
TIDY_TAG_TD
TIDY_TAG_TEXTAREA
TIDY_TAG_TFOOT
TIDY_TAG_TH
TIDY_TAG_THEAD
TIDY_TAG_TITLE
TIDY_TAG_TR
TIDY_TAG_TR
TIDY_TAG_TT
TIDY_TAG_U
TIDY_TAG_UL
TIDY_TAG_VAR
TIDY_TAG_WBR
TIDY_TAG_XMP

constante
TIDY_ATTR_UNKNOWN
TIDY_ATTR_ABBR
TIDY_ATTR_ACCEPT
TIDY_ATTR_ACCEPT_CHARSET
TIDY_ATTR_ACCESSKEY
TIDY_ATTR_ACTION
TIDY_ATTR_ADD_DATE
TIDY_ATTR_ALIGN
TIDY_ATTR_ALINK
TIDY_ATTR_ALT
TIDY_ATTR_ARCHIVE

TIDY_ATTR_AXIS
TIDY_ATTR_BACKGROUND
TIDY_ATTR_BGCOLOR
TIDY_ATTR_BGPROPERTIES
TIDY_ATTR_BORDER
TIDY_ATTR_BORDERCOLOR
TIDY_ATTR_BOTTOMMARGIN
TIDY_ATTR_CELLPADDING
TIDY_ATTR_CELLSPACING
TIDY_ATTR_CHAR
TIDY_ATTR_CHAROFF
TIDY_ATTR_CHARSET
TIDY_ATTR_CHECKED
TIDY_ATTR_CITE
TIDY_ATTR_CLASS
TIDY_ATTR_CLASSID
TIDY_ATTR_CLEAR
TIDY_ATTR_CODE
TIDY_ATTR_CODEBASE
TIDY_ATTR_CODETYPE
TIDY_ATTR_COLOR
TIDY_ATTR_COLS
TIDY_ATTR_COLSPAN
TIDY_ATTR_COMPACT
TIDY_ATTR_CONTENT
TIDY_ATTR_COORDS
TIDY_ATTR_DATA
TIDY_ATTR_DATAFLD
TIDY_ATTR_DATAPAGESIZE
TIDY_ATTR_DATASRC
TIDY_ATTR_DATETIME
TIDY_ATTR_DECLARE
TIDY_ATTR_DEFER
TIDY_ATTR_DIR
TIDY_ATTR_DISABLED
TIDY_ATTR_ENCODING
TIDY_ATTR_ENCTYPE
TIDY_ATTR_FACE
TIDY_ATTR_FOR
TIDY_ATTR_FRAME
TIDY_ATTR_FRAMEBORDER
TIDY_ATTR_FRAMESPACING
TIDY_ATTR_GRIDX
TIDY_ATTR_GRIDY
TIDY_ATTR_HEADERS

TIDY_ATTR_HEIGHT
TIDY_ATTR_HREF
TIDY_ATTR_HREFLANG
TIDY_ATTR_HSPACE
TIDY_ATTR_HTTP_EQUIV
TIDY_ATTR_ID
TIDY_ATTR_ISMAP
TIDY_ATTR_LABEL
TIDY_ATTR_LANG
TIDY_ATTR_LANGUAGE
TIDY_ATTR_LAST_MODIFIED
TIDY_ATTR_LAST_VISIT
TIDY_ATTR_LEFTMARGIN
TIDY_ATTR_LINK
TIDY_ATTR_LONGDESC
TIDY_ATTR_LOWSRC
TIDY_ATTR_MARGINHEIGHT
TIDY_ATTR_MARGINWIDTH
TIDY_ATTR_MAXLENGTH
TIDY_ATTR_MEDIA
TIDY_ATTR_METHOD
TIDY_ATTR_MULTIPLE
TIDY_ATTR_NAME
TIDY_ATTR_NOHREF
TIDY_ATTR_NORESIZE
TIDY_ATTR_NOSHADE
TIDY_ATTR_NOWRAP
TIDY_ATTR_OBJECT
TIDY_ATTR_OnAFTERUPDATE
TIDY_ATTR_OnBEFOREUNLOAD
TIDY_ATTR_OnBEFOREUPDATE
TIDY_ATTR_OnBLUR
TIDY_ATTR_OnCHANGE
TIDY_ATTR_OnCLICK
TIDY_ATTR_OnDATAAVAILABLE
TIDY_ATTR_OnDATASETCHANGED
TIDY_ATTR_OnDATASETCOMPLETE
TIDY_ATTR_OnDBLCLICK
TIDY_ATTR_OnERRORUPDATE
TIDY_ATTR_OnFOCUS
TIDY_ATTR_OnKEYDOWN
TIDY_ATTR_OnKEYPRESS
TIDY_ATTR_OnKEYUP
TIDY_ATTR_OnLOAD
TIDY_ATTR_OnMOUSEDOWN

TIDY_ATTR_OnMOUSEMOVE
TIDY_ATTR_OnMOUSEOUT
TIDY_ATTR_OnMOUSEOVER
TIDY_ATTR_OnMOUSEUP
TIDY_ATTR_OnRESET
TIDY_ATTR_OnROWENTER
TIDY_ATTR_OnROWEXIT
TIDY_ATTR_OnSELECT
TIDY_ATTR_OnSUBMIT
TIDY_ATTR_OnUNLOAD
TIDY_ATTR_PROFILE
TIDY_ATTR_PROMPT
TIDY_ATTR_RBSPAN
TIDY_ATTR_READONLY
TIDY_ATTR_REL
TIDY_ATTR_REV
TIDY_ATTR_RIGHTMARGIN
TIDY_ATTR_ROWS
TIDY_ATTR_ROWSPAN
TIDY_ATTR_RULES
TIDY_ATTR_SCHEME
TIDY_ATTR_SCOPE
TIDY_ATTR_SCROLLING
TIDY_ATTR_SELECTED
TIDY_ATTR_SHAPE
TIDY_ATTR_SHOWGRID
TIDY_ATTR_SHOWGRIDX
TIDY_ATTR_SHOWGRIDY
TIDY_ATTR_SIZE
TIDY_ATTR_SPAN
TIDY_ATTR_SRC
TIDY_ATTR_STANDBY
TIDY_ATTR_START
TIDY_ATTR_STYLE
TIDY_ATTR_SUMMARY
TIDY_ATTR_TABINDEX
TIDY_ATTR_TARGET
TIDY_ATTR_TEXT
TIDY_ATTR_TITLE
TIDY_ATTR_TOPMARGIN
TIDY_ATTR_TYPE
TIDY_ATTR_USEMAP
TIDY_ATTR_VALIGN
TIDY_ATTR_VALUE
TIDY_ATTR_VALUETYPE

TIDY_ATTR_VERSION
TIDY_ATTR_VLINK
TIDY_ATTR_VSPACE
TIDY_ATTR_WIDTH
TIDY_ATTR_WRAP
TIDY_ATTR_XML_LANG
TIDY_ATTR_XML_SPACE
TIDY_ATTR_XMLNS

constante	description
TIDY_NODETYPE_ROOT	noeud racine
TIDY_NODETYPE_DOCTYPE	doctype
TIDY_NODETYPE_COMMENT	commentaire HTML
TIDY_NODETYPE_PROCINS	Instruction de processus
TIDY_NODETYPE_TEXT	Texte
TIDY_NODETYPE_START	début de tag
TIDY_NODETYPE_END	fin de tag
TIDY_NODETYPE_STARTEND	tag vide
TIDY_NODETYPE_CDATA	CDATA
TIDY_NODETYPE_SECTION	section XML
TIDY_NODETYPE_ASP	code ASP
TIDY_NODETYPE_JSTE	code JSTE
TIDY_NODETYPE_PHP	code PHP
TIDY_NODETYPE_XMLDECL	Déclaration XML

8.131.8 Exemples

Ce simple exemple montre l'utilisation de base de Tidy.

Utilisation de base de Tidy
<pre> <?php ob_start(); ?> <html>Un document html</html> <? \$html = ob_get_clean(); // Specify configuration \$config = array('indent' => true, 'output-xhtml' => true, 'wrap' => 200); // Tidy \$tidy = new tidy; \$tidy->parseString(\$html, \$config, 'utf8'); \$tidy->cleanRepair(); // Output echo \$tidy; ?> </pre>

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Classes pré-définies](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [ob_tidyhandler](#)
- [tidy_access_count](#)
- [tidy_clean_repair](#)
- [tidy_config_count](#)
- [tidy::__construct](#)
- [tidy_diagnose](#)
- [tidy_error_count](#)
- [tidy_get_body](#)
- [tidy_get_config](#)
- [tidy_get_error_buffer](#)
- [tidy_get_head](#)
- [tidy_get_html_ver](#)
- [tidy_get_html](#)
- [tidy_get_opt_doc](#)
- [tidy_get_output](#)
- [tidy_get_release](#)
- [tidy_get_root](#)
- [tidy_get_status](#)
- [tidy_getopt](#)
- [tidy_is_xhtml](#)
- [tidy_is_xml](#)
- [tidy_load_config](#)
- [tidy_node->get_attr](#)
- [tidy_node->get_nodes](#)
- [tidy_node->next](#)
- [tidy_node->prev](#)
- [tidy_parse_file](#)
- [tidy_parse_string](#)
- [tidy_repair_file](#)
- [tidy_repair_string](#)
- [tidy_reset_config](#)
- [tidy_save_config](#)
- [tidy_set_encoding](#)
- [tidy_setopt](#)
- [tidy_warning_count](#)
- [tidyNode->hasChildren](#)
- [tidyNode->hasSiblings](#)
- [tidyNode->isAsp](#)
- [tidyNode->isComment](#)
- [tidyNode->isHtml](#)
- [tidyNode->isJste](#)
- [tidyNode->isPhp](#)
- [tidyNode->isText](#)

8.131.10 tidy_access_count() : Retourne le nombre d'alertes d'accessibilité Tidy rencontrées dans le document

int **tidy_access_count** (tidy object)

tidy_access_count retourne le nombre d'alertes d'accessibilité trouvées dans le document spécifié.

Note

Du fait du comportement de la TidyLib, vous devez appeler tidy_diagnose avant tidy_access_count sinon la fonction retournera toujours 0 . Vous devez également activer l'option `accessibility-check` .

Exemple avec tidy_access_count

```
<?php
$html = '<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html><head><title>Title</title></head>
<body>

<p></p>

</body></html>';

// sélectionner le degré d'accessibilité : 1, 2 or 3
$config = array('accessibility-check' => 3);

$tidy = new tidy();
$tidy->parseString($html, $config);
$tidy->CleanRepair();

/* N'oubliez jamais d'appeler ceci ! */
$tidy->diagnose();

echo tidy_access_count($tidy); //5

?>
```

Voir aussi tidy_error_count et tidy_warning_count .

8.131.11 tidy_clean_repair() : Effectue les opérations de nettoyage et de réparation préparées pour un fichier HTML

Style procédural

bool **tidy_clean_repair** (tidy object)

Style orienté objet (méthode)

bool **tidy->cleanRepair** ()

Cette fonction nettoie et répare l'objet object Tidy passé en argument.

Exemple avec tidy_clean_repair

```
<?php
$html = '<p>test</I>';

$tidy = tidy_parse_string($html);
```

```

tidy_clean_repair($tidy);

echo $tidy;
?>

```

L'exemple ci-dessus va afficher :

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
<head>
<title></title>
</head>
<body>
<p>test</p>
</body>
</html>

```

Voir aussi [tidy_repair_file](#) et [tidy_repair_string](#) .

8.131.12 tidy_config_count() : Retourne le nombre d'erreurs de configuration Tidy rencontrées dans le document

int **tidy_config_count** (tidy object)

[tidy_config_count](#) retourne le nombre d'erreurs rencontrées dans la configuration de l'objet object Tidy spécifié.

Exemple avec [tidy_config_count](#)

```

<?php
$html = '<p>test</I>';

$config = array('doctype' => 'bogus');

$tidy = tidy_parse_string($html, $config);

/* Affichera 1, car 'bogus' n'est pas un doctype valide */
echo tidy_config_count($tidy);
?>

```

8.131.13 tidy::__construct() : Construit un nouvel objet Tidy

tidy **tidy::__construct** (string filename , mixed config , string encoding , bool use_include_path)

[tidy::__construct](#) construit un nouvel objet Tidy.

Si le paramètre filename est fourni, cette fonction lira également ce fichier et initialisera l'objet avec ce fichier, agissant de la même façon que la fonction [tidy_parse_file](#) .

Le paramètre config peut prendre la forme d'un tableau ou d'une chaîne de caractères. Sous forme de chaîne, il représente le nom du fichier de configuration et sinon, c'est un tableau avec les options de configuration. Lisez <http://tidy.sourceforge.net/docs/quickref.html> pour en savoir plus sur chaque option.

8.131.11 tidy_clean_repair() : Effectue les opérations de nettoyage et de réparation préparées pour un fichier

Le paramètre encoding spécifie le jeu de caractères utilisé pour les documents en entrées et sorties. Les valeurs possibles de encoding sont : ascii, latin0, latin1, raw, utf8, iso2022, mac, win1252, ibm858, utf16, utf16le, utf16be, big5 et shiftjis.

Exemple avec `tidy::__construct`

```
<?php

$html = <<< HTML

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head><title>title</title></head>
<body>
<p>paragraph <bt />
text</p>
</body></html>

HTML;

$tidy = new tidy;
$tidy->parseString($html);

$tidy->CleanRepair();

if ($tidy->errorBuffer) {
    echo "Les erreurs suivantes ont été détectées :\n";
    echo $tidy->errorBuffer;
}

?>
```

L'exemple ci-dessus va afficher :

```
Les erreurs suivantes ont été détectées :
line 8 column 14 - Error: <bt> is not recognized!
line 8 column 14 - Warning: discarding unexpected <bt>
```

Voir aussi [tidy_parse_file](#) et [tidy_parse_string](#) .

8.131.14 `tidy_diagnose()` : Etablit le diagnostic pour le document analysé et réparé

Style procédural

bool **tidy_diagnose** (tidy object)

Style orienté objet (méthode)

bool **tidy->diagnose** ()

`tidy_diagnose` lance un diagnostic sur l'objet object Tidy donné, en ajoutant quelques informations concernant le document dans la pile d'erreurs.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec `tidy_diagnose`

```

<?php

$html = <<< HTML
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<p>paragraph</p>
HTML;

$tidy = tidy_parse_string($html);
$tidy->CleanRepair();

// notez la différence entre les deux sorties
echo tidy_get_error_buffer($tidy) . "\n";

$tidy->diagnose();
echo tidy_get_error_buffer($tidy);

?>

```

L'exemple ci-dessus va afficher :

```

line 5 column 1 - Warning: <p> isn't allowed in <head> elements
line 5 column 1 - Warning: inserting missing 'title' element

```

```

line 5 column 1 - Warning: <p> isn't allowed in <head> elements
line 5 column 1 - Warning: inserting missing 'title' element
Info: Doctype given is "-//W3C//DTD XHTML 1.0 Strict//EN"
Info: Document content looks like XHTML 1.0 Strict
2 warnings, 0 errors were found!

```

Voir aussi [tidy_get_error_buffer](#) .

8.131.15 `tidy_error_count()` : Retourne le nombre d'erreurs Tidy rencontrées dans le document

```
int tidy_error_count ( tidy object )
```

`tidy_error_count` retourne le nombre d'erreurs Tidy rencontrées dans le document spécifié.

Exemple avec `tidy_error_count`

```

<?php
$html = '<p>test</i>
<bogustag>bogus</bogustag>';

$tidy = tidy_parse_string($html);

echo tidy_error_count($tidy) . "\n"; //1

echo $tidy->ErrorBuffer;

?>

```

L'exemple ci-dessus va afficher :

```

1
line 1 column 1 - Warning: missing <!DOCTYPE> declaration
line 1 column 8 - Warning: discarding unexpected </i>
line 2 column 1 - Error: <bogustag> is not recognized!
line 2 column 1 - Warning: discarding unexpected <bogustag>
line 2 column 16 - Warning: discarding unexpected </bogustag>
line 1 column 1 - Warning: inserting missing 'title' element

```

Voir aussi [tidy_access_count](#) et [tidy_warning_count](#) .

8.131.16 tidy_get_body() : Retourne un objet TidyNode, commencé à partir de la balise <body>

Style procédural

tidyNode **tidy_get_body** (tidy object)

Style orienté objet (méthode)

tidyNode **tidy->body** ()

[tidy_get_body](#) retourne un objet TidyNode commençant à partir de la balise <body>.

Exemple avec [tidy_get_body](#)

```

<?php
$html = '
<html>
  <head>
    <title>test</title>
  </head>
  <body>
    <p>paragraph</p>
  </body>
</html>';

$tidy = tidy_parse_string($html);

$body = tidy_get_body($tidy);
echo $body->value;
?>

```

L'exemple ci-dessus va afficher :

```

<body>
<p>paragraph</p>
</body>

```

Note

Cette fonction n'est disponible qu'avec le Zend Engine 2, c'est à dire PHP >= 5.0.0.

Voir aussi [tidy_get_head](#) et [tidy_get_html](#) .

8.131.17 tidy_get_config() : Lit la configuration Tidy courante

Style procédural

array **tidy_get_config** (tidy object)

Style orienté objet (méthode)

array **tidy->getConfig** ()

tidy_get_config retourne un tableau avec les options de configuration utilisées par l'objet object Tidy spécifié.

Pour une explication de chaque option, voyez <http://tidy.sourceforge.net/docs/quickref.html> .

Exemple avec tidy_get_config

```
<?php
$html = '<p>test</p>';
$config = array('indent' => TRUE,
               'output-xhtml' => TRUE,
               'wrap', 200);

$tidy = tidy_parse_string($html, $config);

print_r(tidy_get_config($tidy));
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [indent-spaces] => 2
    [wrap] => 200
    [tab-size] => 8
    [char-encoding] => 1
    [input-encoding] => 3
    [output-encoding] => 1
    [newline] => 1
    [doctype-mode] => 1
    [doctype] =>
    [repeated-attributes] => 1
    [alt-text] =>
    [slide-style] =>
    [error-file] =>
    [output-file] =>
    [write-back] =>
    [markup] => 1
    [show-warnings] => 1
    [quiet] =>
    [indent] => 1
    [hide-endtags] =>
    [input-xml] =>
    [output-xml] => 1
    [output-xhtml] => 1
```

```
[output-html] =>
[add-xml-decl] =>
[uppercase-tags] =>
[uppercase-attributes] =>
[bare] =>
[clean] =>
[logical-emphasis] =>
[drop-proprietary-attributes] =>
[drop-font-tags] =>
[drop-empty-paras] => 1
[fix-bad-comments] => 1
[break-before-br] =>
[split] =>
[numeric-entities] =>
[quote-marks] =>
[quote-nbsp] => 1
[quote-ampersand] => 1
[wrap-attributes] =>
[wrap-script-literals] =>
[wrap-sections] => 1
[wrap-asp] => 1
[wrap-jste] => 1
[wrap-php] => 1
[fix-backslash] => 1
[indent-attributes] =>
[assume-xml-procins] =>
[add-xml-space] =>
[enclose-text] =>
[enclose-block-text] =>
[keep-time] =>
[word-2000] =>
[tidy-mark] =>
[gnu-emacs] =>
[gnu-emacs-file] =>
[literal-attributes] =>
[show-body-only] =>
[fix-uri] => 1
[lower-literals] => 1
[hide-comments] =>
[indent-cdata] =>
[force-output] => 1
[show-errors] => 6
[ascii-chars] => 1
[join-classes] =>
[join-styles] => 1
[escape-cdata] =>
[language] =>
[ncr] => 1
[output-bom] => 2
[replace-color] =>
[css-prefix] =>
[new-inline-tags] =>
[new-blocklevel-tags] =>
[new-empty-tags] =>
```

```
[new-pre-tags] =>
[accessibility-check] => 0
[vertical-space] =>
[punctuation-wrap] =>
[merge-divs] => 1
)
```

Voir aussi [tidy_reset_config](#) et [tidy_save_config](#) .

8.131.18 tidy_get_error_buffer() : Retourne les alertes et erreurs qui sont survenues lors de l'analyse du document

Style procédural

string **tidy_get_error_buffer** (tidy object)

Style orienté objet (propriété) :

tidy string errorBuffer

[tidy_get_error_buffer](#) retourne les erreurs et les avertissements rencontrés lors de l'analyse du document spécifié.

Exemple avec [tidy_get_error_buffer](#)

```
<?php
$html = '<p>paragraph</p>';

$tidy = tidy_parse_string($html);
echo tidy_get_error_buffer($tidy);
?>
```

L'exemple ci-dessus va afficher :

```
line 1 column 1 - Warning: missing <!DOCTYPE> declaration
line 1 column 1 - Warning: inserting missing 'title' element
```

Voir aussi [tidy_access_count](#) , [tidy_error_count](#) et [tidy_warning_count](#) .

8.131.19 tidy_get_head() : Retourne un objet TidyNode à partir de la balise <head>

Style procédural

tidyNode **tidy_get_head** (tidy object)

Style orienté objet (méthode)

tidyNode **tidy->head** ()

[tidy_get_head](#) retourne un objet tidyNode à partir de la balise <head>.

Exemple avec [tidy_get_head](#)

```

<?php
$html = '
<html>
  <head>
    <title>test</title>
  </head>
  <body>
    <p>paragraph</p>
  </body>
</html>';

$tidy = tidy_parse_string($html);

$head = tidy_get_head($tidy);
echo $head->value;
?>

```

L'exemple ci-dessus va afficher :

```

<head>
<title>test</title>
</head>

```

Note

Cette fonction n'est disponible qu'avec le Zend Engine 2, c'est à dire PHP >= 5.0.0.

Voir aussi [tidy_get_body](#) et [tidy_get_html](#) .

8.131.20 tidy_get_html_ver() : Détecte le version du code HTML utilisée dans un document

Style procédural

int **tidy_get_html_ver** (tidy object)

Style orienté objet (méthode)

int **tidy->getHtmlVer** ()

[tidy_get_html_ver](#) retourne la version détectée du code HTML pour l'objet object Tidy spécifié.

Attention

[tidy_get_html_ver](#) n'est pas encore implémentée dans la bibliothèque Tidylib elle-même ; de ce fait, elle retournera toujours 0 .

8.131.21 tidy_get_html() : Retourne un objet TidyNode commençant à la balise <html>

Style procédural

TidyNode **tidy_get_html** (tidy object)

Style orienté objet (méthode)

tidyNode **tidy->html** ()

tidy_get_html retourne un objet tidyNode commençant à la balise <html>.

Exemple avec tidy_get_html

```
<?php
$html = '
<html>
  <head>
    <title>test</title>
  </head>
  <body>
    <p>paragraph</p>
  </body>
</html>';

$tidy = tidy_parse_string($html);

$html = tidy_get_html($tidy);
echo $html->value;
?>
```

L'exemple ci-dessus va afficher :

```
<html>
<head>
<title>test</title>
</head>
<body>
<p>paragraph</p>
</body>
</html>
```

Note

Cette fonction n'est disponible qu'avec le Zend Engine 2, c'est à dire PHP >= 5.0.0.

Voir aussi tidy_get_body et tidy_get_head .

8.131.22 tidy_get_opt_doc() : Retourne la documentation pour le nom de l'option donnée

Style procédural

string **tidy_get_opt_doc** (tidy object , string optname)

Style orienté objet (méthode)

string **tidy->getOptDoc** (string optname)

tidy_get_opt_doc retourne la documentation pour le nom de l'option donnée.

Note

Vous devez posséder au moins libtidy datant du 25 avril 2005 pour que cette fonction soit disponible.

8.131.23 tidy_get_output() : Retourne une chaîne représentant les balises telles qu'analysées par Tidy

string **tidy_get_output** (tidy object)

tidy_get_output retourne une chaîne avec le HTML réparé.

Exemple avec tidy_get_output

```
<?php
$html = '<p>paragraphe</i>';
$tidy = tidy_parse_string($html);
$tidy->CleanRepair();
echo tidy_get_output($tidy);
?>
```

L'exemple ci-dessus va afficher :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
<head>
<title></title>
</head>
<body>
<p>paragraphe</p>
</body>
</html>
```

8.131.24 tidy_get_release() : Retourne la date de publication (version) de la bibliothèque Tidy

Style procédural

string **tidy_get_release** (void)

Style orienté objet (méthode)

string **tidy->getRelease** ()

tidy_get_release retourne une chaîne avec la date de publication de la bibliothèque tidy.

8.131.25 tidy_get_root() : Retourne un objet tidyNode représentant la racine du document HTML

Style procédural

tidyNode **tidy_get_root** (tidy object)

Style orienté objet (méthode)

tidyNode **tidy->root** ()

tidy_get_root retourne un objet tidyNode représentant la racine de l'arbre Tidy analysé.

Affichage des noeuds

```
<?php

$html = <<< HTML
<html><body>

<p>paragraphe</p>
<br/>

</body></html>
HTML;

$tidy = tidy_parse_string($html);
dump_nodes($tidy->root(), 1);

function dump_nodes($node, $indent) {

    if($node->hasChildren()) {
        foreach($node->child as $child) {
            echo str_repeat('.', $indent*2) . ($child->name ? $child->name : ''.$child->value.' ');
            dump_nodes($child, $indent+1);
        }
    }
}
?>
```

L'exemple ci-dessus va afficher :

```
..html
....head
.....title
....body
.....p
....."paragraphe"
.....br
```

Note

Cette fonction n'est disponible qu'avec le Zend Engine 2, c'est à dire PHP >= 5.0.0.

8.131.26 tidy_get_status() : Retourne le statut du document spécifié

Style procédural

int **tidy_get_status** (tidy object)

Style orienté objet (méthode)

int **tidy->getStatus** ()

tidy_get_status retourne le statut de l'objet object Tidy spécifié. Elle retourne 0 si aucune erreur/avertissement n'a été rencontrée, 1 pour les avertissements d'accessibilité, et 2 pour les

8.131.25 tidy_get_root() : Retourne un objet tidyNode représentant la racine du document HTML

erreurs.

Exemple avec `tidy_get_status`

```
<?php
$html = '<p>paragraphe</i>';
$tidy = tidy_parse_string($html);

$html2 = '<bogus>test</bogus>';
$tidy2 = tidy_parse_string($html2);

echo tidy_get_status($tidy); //1

echo tidy_get_status($tidy2); //2
?>
```

8.131.27 `tidy_getopt()` : Retourne la valeur de l'option de configuration Tidy

Style procédural

mixed `tidy_getopt` (tidy object , string option)

Style orienté objet (méthode)

mixed `tidy->getOpt` (string option)

`tidy_getopt` retourne la valeur de l'option spécifiée option pour l'objet object . Le type retourné dépend du type de l'option spécifiée option . Vous pouvez trouver une liste de chaque option de configuration ainsi que leur type à : <http://tidy.sourceforge.net/docs/quickref.html> .

Exemple avec `tidy_getopt`

```
<?php

$html = '<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html><head><title>Title</title></head>
<body>

<p></p>

</body></html>';

$config = array('accessibility-check' => 3,
               'alt-text' => 'du texte');

$tidy = new tidy();
$tidy->parseString($html, $config);

var_dump($tidy->getOpt('accessibility-check')); //integer
var_dump($tidy->getOpt('lower-literals')); //boolean
var_dump($tidy->getOpt('alt-text')); //string

?>
```

L'exemple ci-dessus va afficher :

```
int(3)
bool(true)
string(9) "some text"
```

8.131.28 tidy_is_xhtml() : Indique si le document est un document XHTML

Style procédural

```
bool tidy_is_xhtml ( tidy object )
```

Style orienté objet (méthode)

```
bool tidy->isXhtml ( )
```

`tidy_is_xhtml` retourne TRUE si l'objet object Tidy est un document XHTML ou FALSE sinon.

Attention

Cette fonction n'est pas encore implémentée par la bibliothèque Tidylib elle-même, donc elle retournera toujours FALSE .

8.131.29 tidy_is_xml() : Indique si le document est un document XML générique (non HTML/XHTML)

Style procédural

```
bool tidy_is_xml ( tidy object )
```

Style orienté objet (méthode)

```
bool tidy->isXml ( )
```

`tidy_is_xml` retourne TRUE si l'objet object Tidy est un document XML générique (non HTML/XHTML) ou FALSE sinon.

Attention

Cette fonction n'est pas encore implémentée par la bibliothèque Tidylib elle-même, donc elle retournera toujours FALSE .

8.131.30 tidy_load_config() : Charge un fichier de configuration ASCII Tidy avec l'encodage spécifié

```
void tidy_load_config ( string filename , string encoding )
```

`tidy_load_config` charge le fichier de configuration Tidy avec le jeu de caractères encoding .

Note

Cette fonction n'est disponible qu'avec Tidy 1.0. Elle a été abandonnée en Tidy 2.0 et a donc été supprimée.

8.131.31 tidy_node->get_attr() : Retourne la valeur de l'attribut spécifié

tidy_attr tidy_node->get_attr (int attrib_id)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.131.32 tidy_node->get_nodes() : Retourne un tableau avec les noeuds placés sous le noeud courant, avec l'identifiant spécifié

array tidy_node->get_nodes (int node_id)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.131.33 tidy_node->next() : Retourne le prochain frère du noeud courant

tidy_node tidy_node->next (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.131.34 tidy_node->prev() : Retourne le frère précédent de ce noeud

tidy_node tidy_node->prev (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.131.35 tidy_parse_file() : Analyse les balises d'un fichier ou d'une URI

Style procédural

tidy tidy_parse_file (string filename , mixed config , string encoding , bool use_include_path)

Style orienté objet (méthode)

bool tidy->parseFile (string filename , mixed config , string encoding , bool use_include_path)

tidy_parse_file analyse le fichier spécifié.

Le paramètre config peut prendre la forme d'un tableau ou d'une chaîne de caractères. Sous forme de chaîne, il représente le nom du fichier de configuration et sinon, c'est un tableau avec les options de configuration. Lisez <http://tidy.sourceforge.net/docs/quickref.html> pour en savoir plus sur chaque option.

Le paramètre encoding spécifie le jeu de caractères utilisé pour les documents en entrées et sorties. Les valeurs possibles de encoding sont : ascii, latin0, latin1, raw, utf8, iso2022, mac, win1252, ibm858, utf16, utf16le, utf16be, big5 et shiftjis.

Exemple avec `tidy_parse_file`

```
<?php
$tidy = tidy_parse_file('file.html');

$tidy->cleanRepair();

if(!empty($tidy->error_buf)) {
    echo 'Les erreurs et avertissements suivants ont été rencontrés :'. "\n";
    echo $tidy->error_buf;
}
?>
```

Note

Les paramètres optionnels `config` et `encoding` ont été ajoutés en Tidy 2.0.

Voir aussi [tidy_parse_string](#) , [tidy_repair_file](#) et [tidy_repair_string](#) .

8.131.36 `tidy_parse_string()` : Analyse un document HTML contenu dans une chaîne

Style procédural

`tidy` **`tidy_parse_string`** (`string input` , *mixed config* , *string encoding*)

Style orienté objet (méthode)

`bool` **`tidy->parseString`** (`string input` , *mixed config* , *string encoding*)

[tidy_parse_string](#) analyse un document contenu dans une chaîne.

Le paramètre `config` peut prendre la forme d'un tableau ou d'une chaîne de caractères. Sous forme de chaîne, il représente le nom du fichier de configuration et sinon, c'est un tableau avec les options de configuration. Lisez <http://tidy.sourceforge.net/docs/quickref.html> pour en savoir plus sur chaque option.

Le paramètre `encoding` spécifie le jeu de caractères utilisé pour les documents en entrées et sorties. Les valeurs possibles de `encoding` sont : `ascii`, `latin0`, `latin1`, `raw`, `utf8`, `iso2022`, `mac`, `win1252`, `ibm858`, `utf16`, `utf16le`, `utf16be`, `big5` et `shiftjis`.

Exemple avec `tidy_parse_string`

```
<?php
ob_start();
?>

<html>
  <head>
    <title>test</title>
  </head>
  <body>
    <p>erreur<br />une autre ligne</p>
  </body>
</html>

<?php

$buffer = ob_get_clean();
$config = array('indent' => TRUE,
               'output-xhtml' => TRUE,
```

```

        'wrap', 200);

$tidy = tidy_parse_string($buffer, $config, 'UTF8');

$tidy->cleanRepair();

echo $tidy;
?>

```

L'exemple ci-dessus va afficher :

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>
test
</title>
</head>
<body>
<p>
error<br />
another line
</p>
</body>
</html>

```

Note

Les paramètres optionnels config et encoding ont été ajoutés en Tidy 2.0.

Voir aussi [tidy_parse_file](#) , [tidy_repair_file](#) et [tidy_repair_string](#) .

8.131.37 tidy_repair_file() : Répare un fichier et le renvoie en tant que chaîne

string **tidy_repair_file** (string filename , mixed config , string encoding , bool use_include_path)

[tidy_repair_file](#) répare le fichier fourni et le renvoie sous la forme d'une chaîne.

Exemple avec [tidy_repair_file](#)

```

<?php
$file = 'file.html';

$repaired = tidy_repair_file($file);
rename($file, $file . '.bak');

file_put_contents($file, $repaired);
?>

```

Note

Les paramètres optionnels config et encoding ont été ajoutés en Tidy 2.0.

Voir aussi [tidy_parse_file](#) , [tidy_parse_string](#) et [tidy_repair_string](#) .

8.131.38 tidy_repair_string() : Répare une chaîne HTML en utilisant un fichier de configuration optionnel

string **tidy_repair_string** (string data , *mixed* config , *string* encoding)

tidy_repair_string répare la chaîne fournie.

Exemple avec tidy_repair_string

```
<?php
ob_start ();
?>

<html>
  <head>
    <title>test</title>
  </head>
  <body>
    <p>error</i>
  </body>
</html>

<?php

$buffer = ob_get_clean();
$tidy = tidy_repair_string($buffer);

echo $tidy;
?>
```

L'exemple ci-dessus va afficher :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
<head>
<title>test</title>
</head>
<body>
<p>error</p>
</body>
</html>
```

Note

Les paramètres optionnels config et encoding ont été ajoutés en Tidy 2.0.

Voir aussi tidy_parse_file , tidy_parse_string et tidy_repair_file .

8.131.39 tidy_reset_config() : Redonne les valeurs de configuration par défaut de Tidy

bool **tidy_reset_config** (void)

tidy_reset_config restaure la configuration par défaut de Tidy.

Note

Cette fonction n'est disponible qu'avec Tidy 1.0. Elle a été abandonnée en Tidy 2.0 et a donc été supprimée.

8.131.40 `tidy_save_config()` : Sauve la configuration courante dans un fichier

bool `tidy_save_config` (string filename)

`tidy_save_config` sauve la configuration courante dans le fichier filename . Seules les valeurs différentes des valeurs par défaut sont sauveées.

Voir aussi `tidy_get_config` , `tidy_getopt` , `tidy_reset_config` et `tidy_setopt` .

Note

Cette fonction n'est disponible qu'avec Tidy 1.0. Elle a été abandonnée en Tidy 2.0 et a donc été supprimée.

8.131.41 `tidy_set_encoding()` : Modifie le jeu de caractères pour les entrées/sorties de l'analyseur Tidy

bool `tidy_set_encoding` (string encoding)

`tidy_set_encoding` configure l'encodage pour les documents en entrée et en sortie. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. La valeurs possibles de encoding sont ascii, latin0, latin1, raw, utf8, iso2022, mac, win1252, ibm858, utf16, utf16le, utf16be, big5 et shiftjis.

Note

Cette fonction n'est disponible qu'avec Tidy 1.0. Elle a été abandonnée en Tidy 2.0 et a donc été supprimée.

8.131.42 `tidy_setopt()` : Modifie la valeur de l'option de configuration Tidy

bool `tidy_setopt` (string option , mixed value)

`tidy_setopt` configure l'option option avec la nouvelle valeur value .

Exemple avec `tidy_setopt`

```
<?php
$html = '<p>test</i>';

$tidy = tidy_parse_string($html);

tidy_setopt('indent', FALSE);
?>
```

Voir aussi `tidy_getopt` , `tidy_get_config` , `tidy_reset_config` et `tidy_save_config` .

Note

Cette fonction n'est disponible qu'avec Tidy 1.0. Elle a été abandonnée en Tidy 2.0 et a donc été supprimée.

8.131.43 tidy_warning_count() : Retourne le nombre d'alertes Tidy rencontrées dans le document spécifié

int tidy_warning_count (tidy object)

tidy_warning_count retourne le nombre d'avertissements tidy rencontrés pour le document spécifié.

Exemple avec <u>tidy_warning_count</u>
<pre><?php \$html = '<p>test</i> <bogustag>bogus</bogustag>'; \$tidy = tidy_parse_string(\$html); echo tidy_error_count(\$tidy) . "\n"; //1 echo tidy_warning_count(\$tidy) . "\n"; //5 ?></pre>

Voir aussi tidy_access_count et tidy_error_count .

8.131.44 tidyNode->hasChildren() : Retourne true si le noeud a des enfants

bool tidyNode->hasChildren (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.
Note
Cette fonction était appelée <u>tidy_node->has_children</u> dans PHP 4/Tidy 1.

8.131.45 tidyNode->hasSiblings() : Retourne true si le noeud a des frères

bool tidyNode->hasSiblings (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.
Note
Cette fonction était appelée <u>tidy_node->has_siblings</u> dans PHP 4/Tidy 1.

8.131.46 tidyNode->isAsp() : Retourne TRUE si ce noeud Tidy est du code ASP

bool tidyNode->isAsp (void)

tidyNode->isAsp retourne TRUE si le noeud courant est du code ASP, FALSE sinon.

Note
Cette foncton était nommée <u>tidy_node->is_asp</u> en PHP 4/Tidy 1.

8.131.47 tidyNode->isComment() : Retourne true si le noeud représente un commentaire

bool tidyNode->isComment (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.
Note
Cette fonction était appelée <code>tidy_node->is_comment</code> dans PHP 4/Tidy 1.

8.131.48 tidyNode->isHtml() : Retourne true si le noeud est une partie d'un document HTML

bool tidyNode->isHtml (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.
Note
Cette fonction était appelée <code>tidy_node->is_html</code> dans PHP 4/Tidy 1.

8.131.49 tidyNode->isJste() : Renvoie true; si ce node est JSTE

bool tidyNode->isJste (void)

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.
Note
Cette fonction était appelée <code>tidy_node->is_jste</code> dans PHP 4/Tidy 1.

8.131.50 tidyNode->isPhp() : Retourne TRUE si ce noeud est en PHP

bool tidyNode->isPhp (void)

`tidyNode->isPhp` retourne TRUE si le noeud courant est du code PHP, FALSE sinon.

Récupération du code PHP depuis un document HTML
<pre> <?php \$html = <<< HTML <html><head> <?php echo '<title>Titre</title>'; ?> </head> <body> <?php echo 'Bonjour le monde !'; ?> </body></html> HTML; \$tidy = tidy_parse_string(\$html); </pre>

```

$num = 0;

get_php($tidy->html());

function get_php($node) {

    // vérifie si le noeud courant est du code PHP
    if($node->isPHP()) {
        echo "\n\n# PHP node #" . ++$GLOBALS['num'] . "\n";
        echo $node->value;
    }

    // vérifie si le noeud courant a des enfants
    if($node->hasChildren()) {
        foreach($node->child as $child) {
            get_php($child);
        }
    }
}

?>

```

L'exemple ci-dessus va afficher :

```

# PHP node #1
<?php echo '<title>Titre</title>'; ?>

# PHP node #2
<?php
echo 'Bonjour le monde !';
?>

```

Note

Cette fonction était nommée `tidy_node->is_php` en PHP 4/Tidy 1.

8.131.51 `tidyNode->isText()` : Retourne true si le noeud représente du texte (aucun marquage)

bool `tidyNode->isText` (void)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Note

Cette fonction était appelée `tidy_node->is_text` dans PHP 4/Tidy 1.

8.132 Analyseur de code PHP

8.132.1 Introduction

Les fonctions du tokenizer fournissent une interface avec l'analyseur interne de PHP, intégré dans le moteur Zend. En utilisant ces fonctions, vous pouvez écrire votre propre système d'analyse des sources PHP, ou bien des outils de modification du code, sans avoir à refaire toute l'analyse lexicale.

Voir aussi l'annexe sur les [tokens](#) .

8.132.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.132.3 Installation

A partir de PHP 4.3.0, ces fonctions sont activées par défaut. Pour les anciennes versions, vous devez compiler PHP avec l'option `--enable-tokenizer` . Vous pouvez désactiver le support de l'extension tokenizer avec l'option `--disable-tokenizer` .

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de bibliothèque supplémentaire pour disposer de ces fonctions.

Note

Le support automatique du tokenizer est disponible depuis PHP 4.3.0.
--

8.132.4 Constantes pré-définies

Lorsque l'extension est compilée dans PHP ou chargée dynamiquement au démarrage, les jetons listés dans la [Liste des token de l'analyseur](#) sont définis comme des constantes.

8.132.5 Exemples

Voici un exemple simple de script PHP qui utilise le tokenizer pour lire un fichier, supprimer tous les commentaires de la source, et afficher uniquement du code.

Suppression de commentaires avec le tokenizer

<pre> <?php /* T_ML_COMMENT n'existe pas en PHP 5. * Les trois lignes suivantes définissent l'ordre * pour préserver une compatibilité ascendante. * * Les deux lignes suivantes la constante T_DOC_COMMENT uniquement présente en PHP 5, * que nous appellerons T_ML_COMMENT pour PHP 4. */ if (!defined('T_ML_COMMENT')) { define('T_ML_COMMENT', T_COMMENT); } else { define('T_DOC_COMMENT', T_ML_COMMENT); </pre>

```
}  
$source = file_get_contents("somefile.php");  
$tokens = token_get_all($source);  
  
foreach ($tokens as $token) {  
    if (is_string($token)) {  
        // simple 1-character token  
        echo $token;  
    } else {  
        // token array  
        list($id, $text) = $token;  
        switch ($id) {  
            case T_COMMENT:  
            case T_ML_COMMENT: // nous avons définie cette constante  
            case T_DOC_COMMENT: // et celle-là  
                // aucune action sur les commentaires  
                break;  
            default:  
                // rien d'autre -> affiche "as is"  
                echo $text;  
                break;  
        }  
    }  
}  
?>
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [token_get_all](#)
- [token_name](#)

8.132.7 token_name() : Lit le nom d'un élément de code source

string **token_name** (int token)

token_name retourne le nom symbolique pour une valeur PHP token value.

8.133 Fonctions ODBC (unifiés)

8.133.1 Introduction

En plus du support de l'ODBC normal, l'ODBC unifié de PHP vous donne accès à diverses bases de données qui ont emprunté la sémantique des API ODBC pour implémenter leur propres API. Au lieu de maintenir de multiples pilotes qui sont similaires, ces pilotes ont été rassemblés dans un jeu de fonctions ODBC uniques.

Les bases de données suivantes sont supportées par l'ODBC unifié : [Adabas D](#) , [IBM DB2](#) , [iODBC](#) , [Solid](#) , et [Sybase SQL Anywhere](#) .

Note

Il n'y a pas d'ODBC utilisé lors des connexions aux bases de données ci-dessus. Les fonctions que vous utiliserez portent des noms évocateurs, et utilisent les mêmes syntaxes que leurs cousines d'ODBC. L'exception à ceci est iODBC. En compilant PHP avec le support iODBC, vous pourrez utiliser n'importe quel pilote compatible ODBC avec vos applications PHP. iODBC est mis à jour sur [OpenLink Software](#) . Plus d'informations sur iODBC, ainsi qu'un HOWTO (en anglais), sont disponibles sur [www.iodbc.org](#) .

8.133.2 Pré-requis

Pour accéder à l'une des bases de données supportées, il faut que les bibliothèques équivalentes soient installées.

8.133.3 Installation

- with-adabas[=DIR]
Inclut le support Adabas D. DIR est le dossier d'installation d'Adabas D. Par défaut, c'est /usr/local .
- with-sapdb[=DIR]
Inclut le support SAP DB. DIR est le dossier d'installation de SAP DB. Par défaut, c'est /usr/local .
- with-solid[=DIR]
Inclut le support Solid. DIR est le dossier d'installation de Solid. Par défaut, c'est /usr/local/solid .
- with-ibm-db2[=DIR]
Inclut le support IBM DB2. DIR est le dossier d'installation de DB2. Par défaut, c'est /home/db2inst1/sqllib .
- with-empres[=DIR]
Inclut le support Empress. DIR est le dossier d'installation de Empress. Par défaut, c'est \$EMPRESSPATH . Depuis PHP 4, cette option ne supporte que Empress Version 8.60 et plus récent.
- with-empres-bcs[=DIR]
Inclut le support Empress Local Access. DIR est le dossier d'installation de Empress. Par défaut, c'est \$EMPRESSPATH . Depuis PHP 4, cette option ne supporte que Empress Version 8.60 et plus récent.
- with-birdstep[=DIR]
Inclut le support Birdstep. DIR est le dossier d'installation de Birdstep. Par défaut, c'est /usr/local/birdstep .
- with-custom-odbc[=DIR]

Inclut le support d'un ODBC défini par l'utilisateur. DIR est le dossier d'installation d'ODBC. Par défaut, c'est /usr/local . Assurez-vous que la variable CUSTOM_ODBC_LIBS est définie, et que le fichier odbc.h est dans votre chemin d'inclusion, c'est-à-dire que vous devriez définir les lignes suivantes pour Sybase SQL Anywhere 5.5.00 sous QNX, avant d'utiliser le script de configuration :
 CPPFLAGS="-DODBC_QNX -DSQLANY_BUG" LDFlags=-linux
 CUSTOM_ODBC_LIBS="-ldblib -lodbc".

--with-iodbc[=DIR]

Inclut le support iODBC. DIR est le dossier d'installation de iODBC. Par défaut, c'est /usr/local .

--with-esoob[=DIR]

Inclut le support Easysoft OOB. DIR est le dossier d'installation de OOB. Par défaut, c'est /usr/local/easysoft/oob/client.

--with-unixodbc[=DIR]

Inclut le support unixODBC. DIR est le dossier d'installation d'unixODBC. Par défaut, c'est /usr/local .

--with-openlink[=DIR]

Inclut le support OpenLink ODBC. DIR est le dossier d'installation OpenLink. Par défaut, c'est /usr/local . C'est le même que pour iODBC.

--with-dbmaker[=DIR]

Inclut le support DBMaker. DIR est le dossier d'installation DBMaker. Par défaut, c'est le chemin de la dernière installation de DBMaker (par exemple /home/dbmaker/3.6).

Pour désactiver le support unifié ODBC dans **PHP 3** ajoutez --disable-unified-odbc dans votre ligne de configuration. Cela n'est possible que si iODBC, Adabas, Solid, Velocis ou une interface ODBC personnalisée est activée.

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de bibliothèque supplémentaire pour disposer de ces fonctions.

8.133.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

Nom	Par défaut	Modifiable	Historique
odbc.default_db *	NULL	PHP_INI_ALL	
odbc.default_user *	NULL	PHP_INI_ALL	
odbc.default_pw *	NULL	PHP_INI_ALL	
odbc.allow_persistent	"1"	PHP_INI_SYSTEM	
odbc.check_persistent	"1"	PHP_INI_SYSTEM	
odbc.max_persistent	"-1"	PHP_INI_SYSTEM	
odbc.max_links	"-1"	PHP_INI_SYSTEM	
odbc.defaultlrl	"4096"	PHP_INI_ALL	
odbc.defaultbinmode	"1"	PHP_INI_ALL	

Note

Les entrées marquées avec une étoile ne sont pas encore implémentées.

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [Directives du php.ini](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

- odbc.default_db chaîne de caractères
La source de données ODBC si aucune n'est spécifiée dans les fonctions [odbc_connect](#) ou [odbc_pconnect](#) .
- odbc.default_user chaîne de caractères
Nom d'utilisateur à utiliser si aucun n'est spécifié dans les fonctions [odbc_connect](#) ou [odbc_pconnect](#) .
- odbc.default_pw chaîne de caractères
Mot de passe à utiliser si aucun n'est spécifié dans les fonctions [odbc_connect](#) ou [odbc_pconnect](#) .
- odbc.allow_persistent booléen
Est-ce que les connexions ODBC persistantes sont autorisées ou pas ?
- odbc.check_persistent booléen
Vérifie que la connexion est valide avant de l'utiliser.
- odbc.max_persistent entier
Nombre maximum de connexions persistantes par processus.
- odbc.max_links entier
Le nombre maximum de connexions ODBC par processus, y compris les connexions persistantes.
- odbc.defaultlrl entier
Gestion des champs de type LONG. Spécifie le nombre d'octets retournés dans les variables.
- Lorsqu'un entier est utilisé, sa valeur est mesuré en octets. Vous pouvez également utiliser la notation sténographique comme décrit dans cette [entrée de la FAQ](#) .
- odbc.defaultbinmode entier
Gestion des données binaires.

8.133.5 Types de ressources

Cette extension ne définit aucune ressource.

8.133.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

ODBC_TYPE ([entier](#))
ODBC_BINMODE_PASSTHRU ([entier](#))
ODBC_BINMODE_RETURN ([entier](#))
ODBC_BINMODE_CONVERT ([entier](#))
SQL_ODBC_CURSORS ([entier](#))
SQL_CUR_USE_DRIVER ([entier](#))
SQL_CUR_USE_IF_NEEDED ([entier](#))
SQL_CUR_USE_ODBC ([entier](#))
SQL_CONCURRENCY ([entier](#))
SQL_CONCUR_READ_ONLY ([entier](#))
SQL_CONCUR_LOCK ([entier](#))
SQL_CONCUR_ROWVER ([entier](#))
SQL_CONCUR_VALUES ([entier](#))
SQL_CURSOR_TYPE ([entier](#))
SQL_CURSOR_FORWARD_ONLY ([entier](#))
SQL_CURSOR_KEYSET_DRIVEN ([entier](#))

SQL_CURSOR_DYNAMIC ([entier](#))
SQL_CURSOR_STATIC ([entier](#))
SQL_KEYSET_SIZE ([entier](#))
SQL_CHAR ([entier](#))
SQL_VARCHAR ([entier](#))
SQL_LONGVARCHAR ([entier](#))
SQL_DECIMAL ([entier](#))
SQL_NUMERIC ([entier](#))
SQL_BIT ([entier](#))
SQL_TINYINT ([entier](#))
SQL_SMALLINT ([entier](#))
SQL_INTEGER ([entier](#))
SQL_BIGINT ([entier](#))
SQL_REAL ([entier](#))
SQL_FLOAT ([entier](#))
SQL_DOUBLE ([entier](#))
SQL_BINARY ([entier](#))
SQL_VARBINARY ([entier](#))
SQL_LONGVARBINARY ([entier](#))
SQL_DATE ([entier](#))
SQL_TIME ([entier](#))
SQL_TIMESTAMP ([entier](#))
SQL_TYPE_DATE ([entier](#))
SQL_TYPE_TIME ([entier](#))
SQL_TYPE_TIMESTAMP ([entier](#))
SQL_BEST_ROWID ([entier](#))
SQL_ROWVER ([entier](#))
SQL_SCOPE_CURROW ([entier](#))
SQL_SCOPE_TRANSACTION ([entier](#))
SQL_SCOPE_SESSION ([entier](#))
SQL_NO_NULLS ([entier](#))
SQL_NULLABLE ([entier](#))
SQL_INDEX_UNIQUE ([entier](#))
SQL_INDEX_ALL ([entier](#))
SQL_ENSURE ([entier](#))
SQL_QUICK ([entier](#))

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [odbc_autocommit](#)
- [odbc_binmode](#)
- [odbc_close_all](#)
- [odbc_close](#)
- [odbc_columnprivileges](#)
- [odbc_columns](#)
- [odbc_commit](#)
- [odbc_connect](#)
- [odbc_cursor](#)
- [odbc_data_source](#)
- [odbc_do](#)

- [odbc_error](#)
- [odbc_errormsg](#)
- [odbc_exec](#)
- [odbc_execute](#)
- [odbc_fetch_array](#)
- [odbc_fetch_into](#)
- [odbc_fetch_object](#)
- [odbc_fetch_row](#)
- [odbc_field_len](#)
- [odbc_field_name](#)
- [odbc_field_num](#)
- [odbc_field_precision](#)
- [odbc_field_scale](#)
- [odbc_field_type](#)
- [odbc_foreignkeys](#)
- [odbc_free_result](#)
- [odbc_gettypeinfo](#)
- [odbc_longreadlen](#)
- [odbc_next_result](#)
- [odbc_num_fields](#)
- [odbc_num_rows](#)
- [odbc_pconnect](#)
- [odbc_prepare](#)
- [odbc_primarykeys](#)
- [odbc_procedurecolumns](#)
- [odbc_procedures](#)
- [odbc_result_all](#)
- [odbc_result](#)
- [odbc_rollback](#)
- [odbc_setoption](#)
- [odbc_specialcolumns](#)
- [odbc_statistics](#)
- [odbc_tableprivileges](#)
- [odbc_tables](#)

8.133.8 `odbc_binmode()` : Modifie la gestion des colonnes de données binaires

bool **odbc_binmode** (resource result_id , int mode)

Types ODBC SQL affectés: BINARY, VARBINARY, LONGVARBINARY.

- ODBC_BINMODE_PASSTHRU: Mode Passthru
- ODBC_BINMODE_RETURN: Retourne tel quel.
- ODBC_BINMODE_CONVERT: Convertit en char et retourne la valeur.

Lorsqu'une donnée SQL est convertie en caractère C, les 8 bits du caractère source sont représentés par deux caractères ASCII. Ces caractères sont des représentations ASCII des nombres au format hexadécimal. Par exemple, le binaire 00000001 est converti en "01" et le binaire 11111111 est converti en "FF" .

mode	longueur	résultat
ODBC_BINMODE_PASSTHRU	0	passthru

ODBC_BINMODE_RETURN	0	passthru
ODBC_BINMODE_CONVERT	0	passthru
ODBC_BINMODE_PASSTHRU	0	passthru
ODBC_BINMODE_PASSTHRU	0	passthru
ODBC_BINMODE_RETURN	0	Tel quel
ODBC_BINMODE_CONVERT	0	Caractère

Si `odbc_fetch_into` est utilisé, `passthru` signifie qu'une chaîne vide sera retournée pour ces colonnes.

Si `result_id` vaut 0 , ces paramètres seront appliqués aux nouveaux résultats.

Note
La valeur par défaut de <code>longreadlen</code> est 4096 et celle de <code>odbc_binmode</code> est <code>ODBC_BINMODE_RETURN</code> . La gestion des colonnes binaires est aussi modifiée par <code>odbc_longreadlen</code> .

8.133.9 `odbc_close_all()` : Ferme toutes les connexions ODBC

void `odbc_close_all` (void)

`odbc_close_all` ferme toutes les connexions ODBC à des sources de données.

Note
<code>odbc_close_all</code> échouera s'il y a des transactions en cours sur cette connexion. Dans ce cas, la connexion restera ouverte.

8.133.10 `odbc_close()` : Ferme une connexion ODBC

void `odbc_close` (resource connection_id)

`odbc_close` ferme la connexion avec la source de données représentée par l'identifiant de connexion `connection_id` .

Note
<code>odbc_close</code> échouera s'il y a des transactions en cours sur cette connexion. Dans ce cas, la connexion restera ouverte.

8.133.11 `odbc_columnprivileges()` : Liste les colonnes et leurs droits associés

resource `odbc_columnprivileges` (resource connection_id , string qualifier , string owner , string table_name , string column_name)

`odbc_columnprivileges` liste les colonnes et leurs droits associés pour la table `table_name` .
`odbc_columnprivileges` retourne un identifiant de résultat ODBC, ou bien `FALSE` en cas d'erreur.

Le résultat possède les colonnes suivantes :

- TABLE_QUALIFIER
- TABLE_OWNER
- TABLE_NAME

- GRANTOR
- GRANTEE
- PRIVILEGE
- IS_GRANTABLE

Le résultat est ordonné par TABLE_QUALIFIER, TABLE_OWNER et TABLE_NAME.

Le paramètre column_name accepte des masques de recherche ('%' pour remplacer zéro ou plus caractères, et '_' pour n'en remplacer qu'un seul).

8.133.12 `odbc_columns()` : Liste les colonnes d'une table

resource **odbc_columns** (resource connection_id , string qualifier , string schema , string table_name , string column_name)

`odbc_columns` liste toutes les colonnes de la source de données. `odbc_columns` retourne un identifiant de résultat ODBC contenant les informations ou bien FALSE en cas d'erreur.

Le résultat possède les colonnes suivantes :

- TABLE_QUALIFIER
- TABLE_SCHEM
- TABLE_NAME
- COLUMN_NAME
- DATA_TYPE
- TYPE_NAME
- PRECISION
- LENGTH
- SCALE
- RADIX
- NULLABLE
- REMARKS

Le résultat est ordonné par TABLE_QUALIFIER, TABLE_SCHEM et TABLE_NAME.

Les paramètres owner , column_name et table_name acceptent des masques de recherche ('%' pour remplacer zéro ou plus caractères, et '_' pour n'en remplacer qu'un seul).

Voir aussi `odbc_columnprivileges` pour connaître les droits associés.

8.133.13 `odbc_commit()` : Valide une transaction ODBC

bool **odbc_commit** (resource connection_id)

`odbc_commit` valide toutes les transactions en cours sur la connexion connection_id . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

8.133.14 `odbc_connect()` : Connexion à une source

resource **odbc_connect** (string dsn , string user , string password , int cursor_type)

odbc_connect retourne un identifiant de connexion ODBC ou 0 (FALSE) en cas d'erreur.

L'identifiant de connexion retourné par cette fonction est nécessaire pour toutes les autres fonctions ODBC. Vous pouvez avoir de multiples connexions en même temps. Le quatrième paramètre fixe le type de pointeur de résultat utilisé pour cette connexion. Ce paramètre n'est généralement pas nécessaire, mais il peut être utile pour contourner certains problèmes ODBC.

Avec certains pilotes ODBC, l'exécution de procédures enregistrées complexes peut produire l'erreur suivante : "Cannot open a cursor on a stored procedure that has anything other than a single select statement in it", ce qui signifie : "Impossible de créer un pointeur de résultat dans une procédure enregistrée qui est réduite à une simple sélection (SELECT)". Utiliser l'option SQL_CUR_USE_ODBC permet d'éviter cette erreur. De plus, certains pilotes ne supportent pas le paramètre optionnel de numéro de ligne dans odbc_fetch_row . SQL_CUR_USE_ODBC peut aussi permettre de résoudre ces problèmes.

Les constantes suivantes sont définies comme types de pointeur :

- SQL_CUR_USE_IF_NEEDED
- SQL_CUR_USE_ODBC
- SQL_CUR_USE_DRIVER
- SQL_CUR_DEFAULT

Pour les connexions persistantes, reportez-vous à odbc_pconnect .

8.133.15 odbc_cursor() : Lecture du pointeur de fiche courante (cursorname)

string **odbc_cursor** (resource result_id)

odbc_cursor lit le pointeur de fiche courante (cursorname) pour le résultat result_id .

8.133.16 odbc_data_source() : Retourne des informations sur la connexion courante

array **odbc_data_source** (resource connection_id , int fetch_type)

odbc_data_source retourne FALSE en cas d'erreur, et un tableau en cas de réussite.

odbc_data_source retourne une liste de DNS disponibles (après l'avoir appelée plusieurs fois). Le paramètre connection_id est une connexion ODBC valide. Le paramètre fetch_type peut être l'une des deux constantes suivantes : SQL_FETCH_FIRST ou SQL_FETCH_NEXT . Utilisez SQL_FETCH_FIRST la première fois que la fonction est appelée, puis SQL_FETCH_NEXT .

8.133.17 odbc_do() : Synonyme de odbc_exec

resource **odbc_do** (resource conn_id , string query)

odbc_do exécute la requête query sur la connexion connection_id .

8.133.18 `odbc_error()` : Lit le dernier code d'erreur

string `odbc_error` (*resource connection_id*)

`odbc_error` retourne un état ODBC sur 6 chiffres, ou une chaîne vide s'il n'y avait plus d'erreurs. Si `connection_id` est spécifié, le dernier état ODBC de cette connexion est retourné. Si `connection_id` est omis, c'est le dernier état de n'importe quelle connexion qui est retourné.

Cette fonction retourne beaucoup de valeurs significatives uniquement si la dernière requête `odbc_exec` échoue (i.e. `odbc_exec` retourne FALSE).

Voir aussi `odbc_errormsg` et `odbc_exec` .

8.133.19 `odbc_errormsg()` : Lit le dernier message d'erreur

string `odbc_errormsg` (*resource connection_id*)

`odbc_errormsg` retourne une chaîne contenant le dernier message d'erreur ODBC, ou une chaîne vide s'il n'y avait pas d'erreur. Si `connection_id` est spécifié, le dernier état ODBC de cette connexion est retourné. Si `connection_id` est omis, c'est le dernier état de n'importe quelle connexion qui est retourné.

Cette fonction retourne beaucoup de valeurs significatives uniquement si la dernière requête `odbc_exec` échoue (i.e. `odbc_exec` retourne FALSE).

Voir aussi `odbc_error` et `odbc_exec` .

8.133.20 `odbc_exec()` : Prépare et exécute une requête SQL

resource `odbc_exec` (*resource connection_id* , string *query_string* , *int flags*)

`odbc_exec` retourne FALSE en cas d'erreur, ou bien retourne un identifiant de résultat ODBC en cas d'exécution réussie.

`odbc_exec` envoie une commande SQL à la source de données représentée par `connection_id` . Ce paramètre doit être un identifiant valide de connexion, retourné par `odbc_connect` ou `odbc_pconnect` .

Voir aussi `odbc_prepare` et `odbc_execute` pour les exécutions multiples de requêtes SQL.

8.133.21 `odbc_execute()` : Exécute une requête SQL préparée

bool `odbc_execute` (*resource result_id* , *array parameters_array*)

`odbc_execute` exécute une requête SQL préparée par `odbc_prepare` . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. Le tableau de paramètres `parameters_array` ne sert que si vous avez besoin de paramétrer votre requête.

Les valeurs du paramètre `parameter_array` seront substituées dans les variables de requêtes de la requête préparée.

Tout paramètre de `parameter_array` qui commence et termine par des guillemets simples sera considéré comme un nom de fichier à lire et envoyé à la base de données, avec la variable de requête appropriée.

Note

Depuis PHP 4.1.1, cette fonction de lecture de fichier a les limitations suivantes :

- La lecture du fichier **n'est** sujette à aucune restriction de la page du safe mode ou de la directive open-basedir . Cela a été corrigé en PHP 4.2.0.
- Les fichiers distants ne sont pas supportés.
- Si vous voulez stocker une chaîne de caractères qui commence et se finit réellement par des guillemets, vous devez les protéger avec des antislash, ou bien en ajoutant un espace au début ou à la fin de la chaîne, pour éviter que ce paramètre soit confondu avec un nom de fichier. Si ce n'est pas possible dans le cadre de votre application, vous devrez passer par la fonction odbc_exec .

8.133.22 `odbc_fetch_array()` : Lit une ligne de résultat dans un tableau associatif

array `odbc_fetch_array` (resource result , int rownumber)

Lit une ligne de résultat dans un tableau associatif depuis une requête ODBC. Voir le changelog ci-dessous pour savoir depuis quand cette fonction est disponible.

8.133.23 `odbc_fetch_into()` : Lit une ligne de résultat, et la place dans un tableau

int `odbc_fetch_into` (resource result_id , array result_array , int rownumber) bool
`odbc_fetch_into` (resource result_id , int rownumber , array result_array)

`odbc_fetch_into` retourne le nombre de colonnes dans le résultat, ou FALSE en cas d'erreur. `result_array` doit avoir été passé par référence, mais il peut être de n'importe quel type, étant donné qu'il sera converti en tableau. Le tableau contiendra les valeurs des colonnes, ces dernières étant numérotées à partir de 0.

Exemple avec `odbc_fetch_into` (avant PHP 4.0.6)

```
<?php
$rc = odbc_fetch_into($res_id, $my_array);
?>
```

ou

```
<?php
$rc = odbc_fetch_into($res_id, $row, $my_array);
$rc = odbc_fetch_into($res_id, 1, $my_array);
?>
```

Jusqu'en PHP 4.0.5, le paramètre `result_array` n'a plus besoin d'être passé par référence.

Depuis PHP 4.0.6, le paramètre `rownumber` ne peut pas être passé comme une constante, mais comme une variable.

Depuis PHP 4.2.0, les paramètres `result_array` et `rownumber` ont été inversés. Cela permet à `rownumber` d'être à nouveau une constante. Cette modification est la dernière pour cette fonction.

Exemple avec `odbc_fetch_into` (après PHP 4.0.6)

```
<?php
$rc = odbc_fetch_into($res_id, $my_array);
?>
```

OU

```
<?php
$row = 1;
$rc = odbc_fetch_into($res_id, $row, $my_array);
?>
```

Exemple avec `odbc_fetch_into` en PHP 4.2.0

```
<?php
$rc = odbc_fetch_into($res_id, $my_array);
?>
```

OU

```
<?php
$rc = odbc_fetch_into($res_id, $my_array, 2);
?>
```

8.133.24 `odbc_fetch_object()` : Lit une ligne de résultat dans un objet

objet `odbc_fetch_object` (resource `result` , int `rownumber`)

Lit une ligne de résultat dans un objet depuis une requête ODBC. Voir le changelog ci-dessous pour savoir depuis quand cette fonction est disponible.

8.133.25 `odbc_fetch_row()` : Lit une ligne de résultat

bool `odbc_fetch_row` (resource `result_id` , int `row_number`)

Si `odbc_fetch_row` a réussi, TRUE est retourné. S'il n'y avait plus de ligne, ou en cas d'erreur, FALSE est retourné.

`odbc_fetch_row` lit une ligne dans le résultat identifié par `result_id` et retourné par `odbc_do` ou `odbc_exec` . Après `odbc_fetch_row` , les champs seront accessibles avec la fonction `odbc_result` .

Si `row_number` est omis, `row_number` va tenter de lire la prochaine ligne dans le résultat. Des appels répétés à `odbc_fetch_row` avec et sans paramètre `row_number` peuvent être combinés librement.

Pour passer en revue toutes les lignes d'un résultat plusieurs fois, vous pouvez appeler `odbc_fetch_row` avec `row_number = 1`, puis continuer à appeler `odbc_fetch_row` sans le paramètre `row_number` pour passer en revue tout le résultat. Si un pilote ne supporte pas la lecture des lignes par numéro, le paramètre sera ignoré.

8.133.26 odbc_field_len() : Lit la longueur d'un champ

int **odbc_field_len** (resource result_id , int field_number)

odbc_field_len retourne la longueur du champ référencé par le nombre field_number , dans la connexion ODBC result_id . Les numéros de champs commencent à 1.

Voir aussi odbc_field_scale pour connaître l'échelle d'un nombre à virgule flottante.

8.133.27 odbc_field_name() : Lit le nom de la colonne

string **odbc_field_name** (resource result_id , int field_number)

odbc_field_name lit le nom de la colonne dont l'index est field_number . La numérotation des colonnes commence à 1. FALSE est retourné en cas d'erreur.

8.133.28 odbc_field_num() : Numéro de colonne

int **odbc_field_num** (resource result_id , string field_name)

odbc_field_num retourne le numéro de la colonne nommée field_name . Ce numéro correspond à l'index du champ dans le résultat ODBC. La numérotation commence à 1. FALSE est retourné en cas d'erreur.

8.133.29 odbc_field_precision() : Alias de odbc_field_len

int **odbc_field_precision** (resource result_id , int field_number)

odbc_field_precision retourne la précision du champ référencé par son numéro field_number , dans le résultat ODBC result_id .

Voir aussi : odbc_field_scale pour connaître l'échelle d'un nombre à virgule flottante.

8.133.30 odbc_field_scale() : Lit l'échelle d'un champ

int **odbc_field_scale** (resource result_id , int field_number)

odbc_field_scale retourne l'échelle du champ référencé par son numéro de champ field_number dans le résultat ODBC result_id .

8.133.31 odbc_field_type() : Type de données d'un champ

string **odbc_field_type** (resource result_id , int field_number)

odbc_field_type retourne le type de données SQL d'un champ, identifié par son index. La numérotation des champs commence à 1.

8.133.32 `odbc_foreignkeys()` : Liste les clés étrangères

resource **`odbc_foreignkeys`** (resource `connection_id` , string `pk_qualifier` , string `pk_owner` , string `pk_table` , string `fk_qualifier` , string `fk_owner` , string `fk_table`)

`odbc_foreignkeys` liste les clés étrangères utilisées dans la table `pk_table` . `odbc_foreignkeys` retourne un identifiant de résultat, ou FALSE en cas d'erreur.

Le résultat possède les colonnes suivantes :

- PKTABLE_QUALIFIER
- PKTABLE_OWNER
- PKTABLE_NAME
- PKCOLUMN_NAME
- FKTABLE_QUALIFIER
- FKTABLE_OWNER
- FKTABLE_NAME
- FKCOLUMN_NAME
- KEY_SEQ
- UPDATE_RULE
- DELETE_RULE
- FK_NAME
- PK_NAME

Si `pk_table` contient un nom de table, `odbc_foreignkeys` retourne la clé primaire de la table `pk_table` , et toutes les clés étrangères qui y font référence.

Si `fk_table` contient un nom de table, `odbc_foreignkeys` retourne la liste des clés étrangères de la table `fk_table` , et les clés primaires (d'autres tables) qui y font référence.

Si `pk_table` et `fk_table` contiennent des noms de tables, `odbc_foreignkeys` retourne la liste des clés étrangères de la table `fk_table` qui utilisent la clé primaire de la table `pk_table` . Cette liste devrait ne contenir qu'une clé au mieux.

8.133.33 `odbc_free_result()` : Libère les ressources associées à un résultat

bool **`odbc_free_result`** (resource `result_id`)

`odbc_free_result` retourne toujours TRUE .

`odbc_free_result` n'est nécessaire que si vous craignez d'utiliser trop de mémoire lors de l'exécution de votre script. Tous les résultats en mémoire seront libérés dès la fin du script. Mais, si vous êtes sûr que vous n'aurez plus besoin d'un résultat jusqu'à la fin de votre script, vous pouvez appeler `odbc_free_result` , et la mémoire associée à `result_id` sera libérée.

Note

Si l'auto-validation est désactivée (voir `odbc_autocommit`) et que vous appelez `odbc_free_result` avant de valider vos requêtes, toutes les transactions préparées seront annulées.

8.133.34 `odbc_gettypeinfo()` : Liste les types de données supportés par une source

resource `odbc_gettypeinfo` (resource `connection_id` , int `data_type`)

`odbc_gettypeinfo` liste les types de données qui sont supportées par une source. `odbc_gettypeinfo` retourne un identifiant de résultat, ou FALSE en cas d'erreur. L'argument optionnel `data_type` peut être utilisé pour restreindre les informations à un seul type de données.

Le résultat possède les colonnes suivantes :

- TYPE_NAME
- DATA_TYPE
- PRECISION
- LITERAL_PREFIX
- LITERAL_SUFFIX
- CREATE_PARAMS
- NULLABLE
- CASE_SENSITIVE
- SEARCHABLE
- UNSIGNED_ATTRIBUTE
- MONEY
- AUTO_INCREMENT
- LOCAL_TYPE_NAME
- MINIMUM_SCALE
- MAXIMUM_SCALE

Le résultat est ordonné par DATA_TYPE et TYPE_NAME.

8.133.35 `odbc_longreadlen()` : Gestion des colonnes de type LONG

bool `odbc_longreadlen` (resource `result_id` , int `length`)

Types ODBC SQL affectés: LONG, LONGVARBINARY.

Le nombre d'octets retournés à PHP est contrôlé par le paramètre `length` . Si sa valeur est 0, les colonnes de type LONG seront transformées en chaînes vides.

Note

La gestion des types LONGVARBINARY est aussi affectée par <code>odbc_binmode</code> .

8.133.36 `odbc_next_result()` : Vérifie si plusieurs résultats sont disponibles

bool `odbc_next_result` (resource `result_id`)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.133.37 `odbc_num_fields()` : Nombre de colonnes dans un résultat

int `odbc_num_fields` (resource result_id)

`odbc_num_fields` retourne le nombre de colonnes dans un résultat ODBC. `odbc_num_fields` retournera -1 en cas d'erreur. L'argument est un identifiant de résultat valide, retourné par `odbc_exec` .

8.133.38 `odbc_num_rows()` : Nombre de lignes dans un résultat

int `odbc_num_rows` (resource result_id)

`odbc_num_rows` retourne le nombre de lignes dans un résultat ODBC. `odbc_num_rows` retournera -1 en cas d'erreur. Pour les commandes INSERT, UPDATE et DELETE, `odbc_num_rows` retourne le nombre de ligne affectées. Pour les commandes SELECT, cela PEUT être le nombre de lignes disponibles, mais ce n'est pas certain.

Note: `odbc_num_rows` après un SELECT retournera -1 avec de nombreux pilotes.

8.133.39 `odbc_pconnect()` : Ouvre une connexion persistante à une source de données

resource `odbc_pconnect` (string dsn , string user , string password , int cursor_type)

`odbc_pconnect` retourne un identifiant de connexion ODBC ou 0 (FALSE) en cas d'erreur. `odbc_pconnect` se comporte de manière similaire à `odbc_connect` , mais la connexion ouverte n'est pas vraiment terminée lorsque le script est achevé. Les prochaines requêtes qui se feront sur une connexion dont les dsn , user , password sont les mêmes que celle-ci (avec `odbc_connect` et `odbc_pconnect`) réutiliseront la connexion ouverte.

Note

Les connexions persistantes n'ont aucun effet si PHP est utilisé comme CGI.

Pour plus de détails sur le paramètre optionnel `cursor_type` , voyez `odbc_connect` . Pour plus de détails sur les connexions persistantes, reportez-vous à la FAQ PHP.

8.133.40 `odbc_prepare()` : Prépare une commande pour l'exécution

resource `odbc_prepare` (resource connection_id , string query_string)

`odbc_prepare` prépare une commande pour l'exécution. Cette fonction retournera FALSE en cas d'erreur.

`odbc_prepare` retourne un identifiant de résultat ODBC si la commande SQL a été préparée avec succès. L'identifiant peut être utilisé plus tard pour exécuter la commande avec `odbc_execute` .

Quelques bases de données (comme IBM DB2, MS SQL Server et Oracle) supportent les procédures stockées qui acceptent les types de paramètres IN, INOUT et OUT comme définis dans les spécifications ODBC. Cependant, le driver unifié ODBC supporte actuellement uniquement le type de paramètres IN pour les procédures stockées.

Dans le code suivant, \$res ne sera valide uniquement si les trois paramètres pour myproc sont des paramètres IN :

```
<?php
$a = 1;
$b = 2;
$c = 3;
$stmt = odbc_prepare($conn, 'CALL myproc(?,?,?)');
$res = odbc_execute($stmt, array($a, $b, $c));
?>
```

Si vous devez appeler une procédure stockée utilisant des paramètres INOUT ou OUT, il est recommandé d'utiliser l'extension native de votre base de données (par exemple, mssql pour MS SQL Server ou oci8 pour Oracle).

8.133.41 odbc_primarykeys() : Liste les colonnes utilisées dans une clé primaire

resource **odbc_primarykeys** (resource connection_id , string qualifier , string owner , string table)

odbc_primarykeys liste les colonnes utilisées dans une clé primaire de la table table .
odbc_primarykeys retourne un identifiant de résultat, ou FALSE en cas d'erreur.

Le résultat possède les colonnes suivantes :

- TABLE_QUALIFIER
- TABLE_OWNER
- TABLE_NAME
- COLUMN_NAME
- KEY_SEQ
- PK_NAME

8.133.42 odbc_procedurecolumns() : Liste les paramètres des procédures

resource **odbc_procedurecolumns** (resource connection_id , string qualifier , string owner , string proc , string column)

odbc_procedurecolumns liste les paramètres d'entrée et de sortie, ainsi que les colonnes utilisées dans les procédures désignées par les paramètres. odbc_procedurecolumns retourne un identifiant de résultat, ou FALSE en cas d'erreur.

Le résultat possède les colonnes suivantes :

- PROCEDURE_QUALIFIER
- PROCEDURE_OWNER
- PROCEDURE_NAME
- COLUMN_NAME
- COLUMN_TYPE
- DATA_TYPE
- TYPE_NAME

- PRECISION
- LENGTH
- SCALE
- RADIX
- NULLABLE
- REMARKS

Le résultat est ordonné par PROCEDURE_QUALIFIER, PROCEDURE_OWNER, PROCEDURE_NAME et COLUMN_TYPE.

Les paramètres owner , proc et column acceptent des masques de recherche ('%' pour remplacer zéro ou plus caractères, et '_' pour n'en remplacer qu'un seul).

8.133.43 `odbc_procedures()` : Liste les procédures stockées

resource **odbc_procedures** (resource connection_id , string qualifier , string owner , string name)

odbc_procedures liste toutes les procédures stockées dans la source de données.

odbc_procedures retourne un identifiant de résultat ODBC contenant les informations ou FALSE en cas d'erreur.

Le résultat possède les colonnes suivantes :

- PROCEDURE_QUALIFIER
- PROCEDURE_OWNER
- PROCEDURE_NAME
- NUM_INPUT_PARAMS
- NUM_OUTPUT_PARAMS
- NUM_RESULT_SETS
- REMARKS
- PROCEDURE_TYPE

Les paramètres owner et name acceptent des masques de recherche ('%' pour remplacer zéro ou plus caractères, et '_' pour n'en remplacer qu'un seul).

8.133.44 `odbc_result_all()` : Affiche le résultat sous la forme d'une table HTML

int **odbc_result_all** (resource result_id , string format)

odbc_result_all retourne le nombre de lignes dans le résultat, ou FALSE en cas d'erreur.

odbc_result_all affiche toutes les lignes d'un résultat. L'affichage se fait au format HTML. Avec l'option format , il est possible de modifier l'aspect global de la table.

8.133.45 `odbc_result()` : Lit un champ de résultat UODBC

mixed **odbc_result** (resource result_id , mixed field)

odbc_result retourne le contenu d'un champ, FALSE si une erreur survient, NULL si le contenu des données est vide ou TRUE pour des données binaires.

field peut être aussi bien un entier, contenant le numéro de colonne du champ, dans le résultat, ou bien une chaîne de caractères, qui représente le nom du champ. Par exemple :

Exemple avec odbc_result

```
<?php
$item_3 = odbc_result($Query_ID, 3 );
$item_val = odbc_result($Query_ID, "val");
?>
```

Le premier appel à odbc_result retourne la valeur du troisième champ de la ligne courante du résultat result_id . Le deuxième appel à odbc_result retourne la valeur du troisième champ dont le nom est "val" de la ligne courante du résultat result_id . Une erreur survient si le paramètre de colonne est inférieur à 1, ou dépasse le nombre de colonnes du résultat. De la même manière, une erreur survient si le nom du champ passé ne correspond à aucun champ dans le résultat.

Les index de champs commencent à 1. Pour plus d'informations sur la façon de lire des colonnes de type binaire ou long, reportez-vous à odbc_binmode et odbc_longreadlen .

8.133.46 odbc_rollback() : Annule une transaction

bool **odbc_rollback** (resource connection_id)

odbc_rollback annule toutes les transactions sur la connexion connection_id . odbc_rollback retourne TRUE en cas de succès, et FALSE en cas d'échec.

8.133.47 odbc_setoption() : Modifie les paramètres ODBC

bool **odbc_setoption** (resource id , int fonction , int option , int param)

odbc_setoption donne accès aux options ODBC pour une connexion particulière ou un résultat de requête. Elle a été écrite pour aider à la résolution de problèmes liés aux pilotes ODBC récalcitrants. Vous aurez sûrement à utiliser odbc_setoption si vous êtes un programmeur ODBC et que vous comprenez les divers effets des options disponibles. Vous aurez aussi besoin d'un bon manuel de référence pour comprendre les options et leur usage. Différentes versions de pilotes supportent différentes versions d'options.

Etant donné que les effets peuvent varier d'un pilote à l'autre, l'utilisation de odbc_setoption dans des scripts voués à être livrés au public est très fortement déconseillée. De plus, certaines options ODBC ne sont pas disponibles car elles doivent être fixées avant l'établissement de la connexion. Cependant, si dans un cas bien spécifique, odbc_setoption vous permet d'utiliser PHP sans que votre patron ne vous pousse à utiliser un produit commercial, alors cela n'a pas d'importance.

id est un identifiant de connexion, ou un identifiant de résultat, pour lequel vous souhaitez modifier des options. Pour SQLSetConnectOption(), c'est un identifiant de connexion. Pour SQLSetStmtOption(), c'est un identifiant de résultat.

fonction est la fonction ODBC à utiliser. La valeur doit être de 1 pour utiliser SQLSetConnectOption() et 2 pour SQLSetStmtOption().

Le paramètre option est l'option à modifier.

Le paramètre param est la valeur de l'option option .

Exemple de modification d'option ODBC

```
<?php
// 1. L'option 102 de SQLSetConnectOption() est SQL_AUTOCOMMIT.
// 1 de SQL_AUTOCOMMIT est SQL_AUTOCOMMIT_ON.
// Cet exemple a le même effet que
// odbc_autocommit($conn, true);

odbc_setoption($conn, 1, 102, 1);

// 2. Option 0 de SQLSetStmtOption() est SQL_QUERY_TIMEOUT.
// Cet exemple fixe le délai d'expiration à 30 secondes.

$result = odbc_prepare($conn, $sql);
odbc_setoption($result, 2, 0, 30);
odbc_execute($result);
?>
```

8.133.48 odbc_specialcolumns() : Retourne l'ensemble optimal de colonnes, qui permet de définir uniquement une ligne dans une table

resource **odbc_specialcolumns** (resource connection_id , int type , string qualifier , string owner , string table , int scope , int nullable)

Lorsque le type est SQL_BEST_ROWID, odbc_specialcolumns retourne la ou les colonnes qui permettent de repérer uniquement chaque ligne d'une table.

Lorsque le type type est SQL_ROWVER, odbc_specialcolumns retourne la colonne ou les colonnes de la table spécifiée, s'il y en a, elles sont automatiquement mises à jour par les données sources lorsque chaque valeur de la ligne est mise à jour par n'importe quelle transaction.

odbc_specialcolumns retourne un identifiant de résultat, ou FALSE en cas d'erreur.

Le résultat possède les colonnes suivantes :

- SCOPE
- COLUMN_NAME
- DATA_TYPE
- TYPE_NAME
- PRECISION
- LENGTH
- SCALE
- PSEUDO_COLUMN

Le résultat est ordonné par SCOPE.

8.133.49 odbc_statistics() : Calcul des statistiques sur une table

resource **odbc_statistics** (resource connection_id , string qualifier , string owner , string table_name , int unique , int accuracy)

odbc_statistics effectue quelques statistiques sur une table et ses index. odbc_statistics retourne un identifiant de résultat, ou FALSE en cas d'erreur.

Le résultat possède les colonnes suivantes :

- TABLE_QUALIFIER
- TABLE_OWNER
- TABLE_NAME
- NON_UNIQUE
- INDEX_QUALIFIER
- INDEX_NAME
- TYPE
- SEQ_IN_INDEX
- COLUMN_NAME
- COLLATION
- CARDINALITY
- PAGES
- FILTER_CONDITION

Le résultat est ordonné par NON_UNIQUE, TYPE, INDEX_QUALIFIER, INDEX_NAME et SEQ_IN_INDEX.

8.133.50 odbc_tableprivileges() : Liste les tables et leurs privilèges

resource **odbc_tableprivileges** (resource connection_id , string qualifier , string owner , string name)

odbc_tableprivileges liste les tables de la source et leurs droits associés. odbc_tableprivileges retourne un identifiant de résultat ODBC, ou bien FALSE en cas d'erreur.

Le résultat possède les colonnes suivantes :

- TABLE_QUALIFIER
- TABLE_OWNER
- TABLE_NAME
- GRANTOR
- GRANTEE
- PRIVILEGE
- IS_GRANTABLE

Le résultat est ordonné par TABLE_QUALIFIER, TABLE_OWNER et TABLE_NAME.

Les paramètres owner et name acceptent des masques de recherche ('%' pour remplacer zéro ou plus caractères, et '_' pour n'en remplacer qu'un seul).

8.133.51 odbc_tables() : Liste les tables d'une source

resource **odbc_tables** (resource connection_id , string qualifier , string owner , string name , string types)

odbc_tables liste toutes les tables de la source et retourne un identifiant de résultat ODBC contenant les informations ou bien FALSE en cas d'erreur.

Le résultat contient les colonnes suivantes :

- TABLE_QUALIFIER
- TABLE_OWNER
- TABLE_NAME
- TABLE_TYPE
- REMARKS

Le résultat est ordonné grâce aux options TABLE_TYPE , TABLE_QUALIFIER , TABLE_OWNER et TABLE_NAME .

Les paramètres owner et name acceptent des masques de recherche ('%' pour remplacer zéro ou plus caractères, et '_' pour n'en remplacer qu'un seul).

Pour supporter les énumérations de qualificateurs propriétaires et types de table, la sémantique suivante pour les paramètres qualifier , owner , name et table_type est disponible :

- Si qualifier est un signe de pourcentage (%), et owner et name sont des chaînes vides, alors le résultat contient la liste des qualifiés valides pour la source (toutes les colonnes hormis TABLE_QUALIFIER contiennent NULL).
- Si owner est un signe de pourcentage (%), et qualifier et name sont des chaînes vides, alors le résultat contient la liste des propriétaires de la source (toutes les colonnes hormis TABLE_OWNER contiennent NULL).
- Si table_type est un signe de pourcentage (%), et qualifier , owner et name sont des chaînes vides, alors le résultat contient la liste des types de tables de la source (toutes les colonnes hormis TABLE_TYPE contiennent NULL).

Si table_type n'est pas une chaîne vide, il doit contenir une liste de valeurs, séparées par des virgules, qui représentent les types recherchés. Chaque valeur peut être insérée entre guillemets simples ('), ou sans guillemets. Par exemple, "'TABLE','VIEW'" ou "TABLE, VIEW". Si la source de données ne supporte pas un type de table donné, odbc_tables ne retournera aucun résultat pour ce type.

Voir aussi odbc_tableprivileges pour connaître les droits associés.

8.134 URL

8.134.1 Introduction

Fonctions permettant de traiter les URL : encoder, décoder, et analyser.

8.134.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.134.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.134.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.134.5 Types de ressources

Cette extension ne définit aucune ressource.

8.134.6 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [base64_decode](#)
- [base64_encode](#)
- [get_headers](#)
- [get_meta_tags](#)
- [http_build_query](#)
- [parse_url](#)
- [rawurldecode](#)
- [rawurlencode](#)
- [urldecode](#)
- [urlencode](#)

8.134.8 base64_encode() : Encode une chaîne en MIME base64

string **base64_encode** (string data)

base64_encode retourne data encodé en base64. Cet encodage est fait pour permettre aux informations binaires d'être manipulées par les systèmes qui ne gèrent pas correctement les 8 bits, comme par exemple, les corps de mail.

Une chaîne encodée base64 prend environs 33% de plus que les données initiales.

Exemple avec base64_encode

```
<?php
$str = 'Ceci est une chaîne encodée';
echo base64_encode($str);
?>
```

L'exemple ci-dessus va afficher :

```
Q2VjaSBlc3QgdW5lIGNoYe5uZSBlbmNvZ0ll
```

Voir aussi base64_decode , chunk_split , convert_uuencode et la RFC 2045 section 6.8.

8.134.9 get_headers() : Retourne tous les en-têtes envoyés par le serveur en réponse à une requête HTTP

array **get_headers** (string url , int format)

get_headers retourne un tableau avec les en-têtes envoyés par le serveur en réponse à une requête HTTP. Retourne FALSE en cas d'échec et une alerte de niveau E_WARNING sera émise.

Si le paramètre optionnel format est définie à 1, get_headers analyse la réponse et définit les index du tableau.

Exemple avec get_headers

```
<?php
$url = 'http://www.example.com';

print_r(get_headers($url));

print_r(get_headers($url, 1));
?>
```

L'exemple ci-dessus va afficher :

```
Array
(
    [0] => HTTP/1.1 200 OK
    [1] => Date: Sat, 29 May 2004 12:28:13 GMT
    [2] => Server: Apache/1.3.27 (Unix) (Red-Hat/Linux)
    [3] => Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
    [4] => ETag: "3f80f-1b6-3e1cb03b"
```

```
[5] => Accept-Ranges: bytes
[6] => Content-Length: 438
[7] => Connection: close
[8] => Content-Type: text/html
)

Array
(
[0] => HTTP/1.1 200 OK
[Date] => Sat, 29 May 2004 12:28:14 GMT
[Server] => Apache/1.3.27 (Unix) (Red-Hat/Linux)
[Last-Modified] => Wed, 08 Jan 2003 23:11:55 GMT
[ETag] => "3f80f-1b6-3e1cb03b"
[Accept-Ranges] => bytes
[Content-Length] => 438
[Connection] => close
[Content-Type] => text/html
)
```

8.134.10 `get_meta_tags()` : Extrait toutes les balises méta d'un fichier HTML

array `get_meta_tags` (string filename , bool use_include_path)

`get_meta_tags` ouvre le fichier filename et l'analyse ligne par ligne à la recherche de balises `<meta>`. Le fichier peut être un fichier local ou une URL . L'analyse cesse lors de la rencontre de la balise `</head>` .

En donnant la valeur de TRUE au paramètre optionnel use_include_path fera que `get_meta_tags` recherchera aussi le fichier dans l'include_path .

Exemple de balise meta

```
<meta name="author" content="name">
<meta name="keywords" content="php documentation">
<meta name="DESCRIPTION" content="a php manual">
<meta name="geo.position" content="49.33;-86.59">
</head>
```

Attention au format des nouvelles lignes : PHP utilise une fonction native pour analyser le fichier, et les fichiers Mac ne seront pas reconnus.

La valeur de la propriété sera utilisé comme clé du tableau, et sa valeur comme valeur correspondante de la clé. Vous pourrez ainsi passer en revue facilement ce tableau avec les fonctions de tableaux standard. Les caractères spéciaux présents dans la valeur seront remplacés par un souligné ("_"), et le reste est converti en minuscule. Si deux balises méta possèdent le même nom, seul la dernière sera retournée.

Résultat de `get_meta_tags`

```
<?php
// Supposons que les balises ci-dessus sont disponibles sur example.com
$tags = get_meta_tags('http://www.example.com/');

// Notez que les clés sont en minuscule, et
// le . a été remplacé par _ dans la clé
echo $tags['author']; // nom
echo $tags['keywords']; // documentation php
echo $tags['description']; // un manuel PHP
```

8.134.9 `get_headers()` : Retourne tous les en-têtes envoyés par le serveur en réponse à une requête HTTP

```
    echo $tags['geo_position']; // 49.33;-86.59
?>
```

Note

Depuis PHP 4.0.5, `get_meta_tags` supporte les attributs HTML sans guillemets.

Voir aussi [htmlentities](#) et [urlencode](#) .

8.134.11 `http_build_query()` : Génère une chaîne de requête en encodage URL

string `http_build_query` (array `formdata` , string `numeric_prefix`)

`http_build_query` génère une chaîne en encodage URL, construite à partir du tableau indexé ou associatif `formdata` . `formdata` peut être un tableau ou un objet contenant une structure mono-dimensionnelle simple, ou un tableau de tableaux (qui peuvent aussi contenir des tableaux). Si des indices numériques sont utilisés dans le tableau de base et que `numeric_prefix` est fourni, il sera utilisé pour préfixer les noms des index pour les éléments du tableau de base seulement. Cela permet de générer des noms de variables valides, si les données sont ensuite décodées par PHP ou une application CGI.

Note

`arg_separator_output` est utilisée pour séparer les arguments.

Utilisation simple de `http_build_query`

```
<?php
$data = array('foo'=>'bar',
             'baz'=>'boom',
             'cow'=>'milk',
             'php'=>'hypertext processor');

echo http_build_query($data); // foo=bar&baz=boom&cow=milk&php=hypertext+processor
?>
```

`http_build_query` avec tableau indexé.

```
<?php
$data = array('foo', 'bar', 'baz', 'boom', 'cow' => 'milk', 'php' =>'hypertext processor');

echo http_build_query($data);
/* affiche :
    0=foo&1=bar&2=baz&3=boom&cow=milk&php=hypertext+processor
*/

echo http_build_query($data, 'myvar_');
/* affiche :
    myvar_0=foo&myvar_1=bar&myvar_2=baz&myvar_3=boom&cow=milk&php=hypertext+processor
*/
?>
```

`http_build_query` avec tableau complexe

```
<?php
$data = array('user'=>array('name'=>'Bob Smith',
                          'age'=>47,
                          'sex'=>'M',
                          'dob'=>'5/12/1956'),
             'pastimes'=>array('golf', 'opera', 'poker', 'rap'),
             'children'=>array('bobby'=>array('age'=>12,
                                             'sex'=>'M'),
```

```

                'sally'=>array('age'=>8,
                               'sex'=>'F')),
            'CEO');
echo http_build_query($data, 'flags_');
?>

```

cet exemple va afficher : (sur plusieurs lignes pour lisibilité)

```

user[name]=Bob+Smith&user[age]=47&user[sex]=M&user[do]b]=5%1F12%1F1956&
pastimes[0]=golf&pastimes[1]=opera&pastimes[2]=poker&pastimes[3]=rap&
children[bobby][age]=12&children[bobby][sex]=M&children[sally][age]=8&
children[sally][sex]=F&flags_0=CEO

```

Note

Seuls les éléments indexés numériquement (" CEO ") dans le tableau de base sont préfixés. Les autres indices numériques à d'autres niveaux n'ont pas besoins d'être préfixés pour avoir des noms valides.

Utilisation de `http_build_query` avec un objet

```

<?php
class myClass {
    var $foo;
    var $baz;

    function myClass() {
        $this->foo = 'bar';
        $this->baz = 'boom';
    }
}

$data = new myClass();

echo http_build_query($data); // foo=bar&baz=boom
?>

```

Voir aussi [parse_str](#) , [parse_url](#) , [urlencode](#) et [array_walk](#) .

8.134.12 `parse_url()` : Analyse une URL et retourne ses composants

array `parse_url` (string url)

Cette fonction analyse une URL et retourne un tableau associatif contenant tous les éléments présents dans l'URL.

Cette fonction n'est **pas** faite pour valider l'URL fournie, elle ne fait que la découper en parties listées ci-dessous. Les URLs partielles sont également acceptées, la fonction [parse_url](#) fera de son mieux pour les analyser correctement.

8.134.13 `rawurldecode()` : Décode une chaîne URL

string `rawurldecode` (string str)

[rawurldecode](#) retourne la chaîne str dont les séquences de caractères %xy , avec xy deux valeurs hexadécimales, auront été remplacées par le caractère ASCII correspondant.

Exemple avec [rawurldecode](#)

```
<?php
echo rawurldecode('foo%20bar%40baz'); // foo bar@baz
?>
```

Note

[rawurldecode](#) ne décode pas le symbole d'addition (" + ") contrairement à la fonction [urldecode](#) . Voir aussi [rawurlencode](#) , [urldecode](#) et [urlencode](#) .

8.134.14 rawurlencode() : Encode une chaîne en URL, selon la RFC 1738

string [rawurlencode](#) (string str)

[rawurlencode](#) retourne une chaîne dont tous les caractères non-alpha-numériques (hormis - _ .) auront été remplacés par des séquences %xy (%), avec xy, deux valeurs hexadécimales. Ce codage est conforme à la RFC 1738 qui évite que les caractères spéciaux soient interprétés comme des délimiteurs, et pour protéger les URL lors du transfert (contrairement à certains systèmes email). Par exemple, si vous voulez mettre un mot de passe dans une URL de FTP :

Exemple avec [rawurlencode](#)

```
echo '<a href="ftp://user:', rawurlencode('foo @+%/'),
      '@ftp.example.com/x.txt">';
```

Ou, si vous transmettez un chemin dans une URL :

Exemple avec [rawurlencode](#)

```
<?php
echo '<a href="http://example.com/department_list_script/',
      rawurlencode('ventes et marketing/Bruxelles'), '>';
?>
```

Voir aussi [rawurldecode](#) , [urldecode](#) et [urlencode](#) , et la [RFC 1738](#) .

8.134.15 urldecode() : Décode une chaîne encodée URL

string [urldecode](#) (string str)

[urldecode](#) décode toutes les séquences % ## et les remplace par leur valeur. La chaîne ainsi décodée est retournée.

Exemple avec [urldecode](#)

```
<?php
$a = explode ('&', $QUERY_STRING);
$i = 0;
while ($i < count ($a)) {
    $b = split ('=', $a[$i]);
    echo 'La valeur du paramètre ', htmlspecialchars(urldecode($b[0])),
          ' est ', htmlspecialchars(urldecode($b[1])), "<br />\n";
    $i++;
}
```

```
?>
```

Voir aussi [urlencode](#) , [rawurlencode](#) et [rawurldecode](#) .

8.134.16 urlencode() : Encode une chaîne en URL

string **urlencode** (string str)

[urlencode](#) retourne une chaîne dont les caractères non alpha-numériques (hormis -_.) sont remplacés par des séquences commençant par un caractère pourcentage (%), suivi de deux chiffres hexadécimaux. Les espaces sont remplacés par des signes plus (+). Ce codage est celui qui est utilisé pour poster des informations dans les formulaires HTML. Le type MIME est application/x-www-form-urlencoded . Ce codage est différent de celui spécifié dans la RFC 1738 (voir [rawurlencode](#)) : pour des raisons historiques, les espaces sont remplacés par des signes plus (+). [urlencode](#) est pratique pour transmettre des informations via une URL. C'est aussi un moyen de passer des informations d'une page à l'autre.

Exemple avec [urlencode](#)

```
<?php
echo '<a href="moncgi?foo=', urlencode ($userinput), '>';
?>
```

Note: Faites bien attention aux variables qui ressemblent à des entités HTML, comme & et © et £ , qui sont analysées par le client web et remplacée par leur valeur. C'est un vrai problème qui a été montré par le W3C depuis longtemps. La référence est ici :

<http://www.w3.org/TR/html4/appendix/notes.html#h-B.2.2> . PHP supporte le remplacement de séparateur d'arguments par un point-virgule, comme recommandé par le W3C, grâce à la directive `arg_separator` du fichier `php.ini` . Malheureusement, la plupart des clients web n'envoient pas leurs données de formulaire avec des points-virgules. Une solution plus portable est d'utiliser & à la place de & comme séparateur. Vous n'avez alors pas à changer la directive `arg_separator` . Laissez-la à & , mais encodez vos URL en utilisant [htmlentities](#) ou [htmlspecialchars](#) .

Exemple avec [urlencode](#) et [htmlentities](#)

```
<?php
$query_string = 'foo=' . urlencode($foo) . '&bar=' . urlencode($bar);
echo '<a href="moncgi?' . htmlentities($query_string) . '>';
?>
```

Voir aussi [urldecode](#) , [htmlentities](#) , [rawurldecode](#) et [rawurlencode](#) .

8.135 Fonctions de gestion des variables

8.135.1 Introduction

Pour plus de détails sur le comportement des variables, reportez-vous à la section [Variables](#) du chapitre [Référence du langage](#) .

8.135.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.135.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.135.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

Nom	Par défaut	Modifiable	Historique
<code>unserialize_callback_func</code>	NULL	PHP_INI_ALL	Disponible depuis PHP 4.2.0.

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [Directives du php.ini](#) .

Voici un éclaircissement sur l'utilisation des directives de configuration.

`unserialize_callback_func` string

La fonction de callback de `unserialize` sera appelée (avec le nom de la classe non définie en paramètre), si PHP ne trouve pas la classe spécifiée, ou si la fonction n'implémente pas la classe manquante. Ne configurez cette option que si vous souhaitez vraiment implémenter une telle fonction.

Voir aussi [unserialize](#) .

8.135.5 Types de ressources

Cette extension ne définit aucune ressource.

8.135.6 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)

- [Constantes pré-définies](#)
- [debug_zval_dump](#)
- [doubleval](#)
- [empty](#)
- [floatval](#)
- [get_defined_vars](#)
- [get_resource_type](#)
- [gettype](#)
- [import_request_variables](#)
- [intval](#)
- [is_array](#)
- [is_bool](#)
- [is_callable](#)
- [is_double](#)
- [is_float](#)
- [is_int](#)
- [is_integer](#)
- [is_long](#)
- [is_null](#)
- [is_numeric](#)
- [is_object](#)
- [is_real](#)
- [is_resource](#)
- [is_scalar](#)
- [is_string](#)
- [isset](#)
- [print_r](#)
- [serialize](#)
- [settype](#)
- [strval](#)
- [unserialize](#)
- [unset](#)
- [var_dump](#)
- [var_export](#)

8.135.8 doubleval() : Alias de [floatval](#)

Cette fonction est un alias de : [floatval](#) .

8.135.9 empty() : Détermine si une variable contient une valeur non nulle

bool **empty** (mixed var)

Détermine si une variable contient une valeur non nulle.

8.135.10 floatval() : Convertit une chaîne en nombre à virgule flottante

float **floatval** (mixed var)

[floatval](#) retourne la valeur de type float (nombre à virgule flottante), extraite à partir du paramètre var .

8.135.11 `get_defined_vars()` : Liste toutes les variables définies

array `get_defined_vars` (void)

`get_defined_vars` retourne un tableau multidimensionnel contenant la liste de toutes les variables définies, qu'elles soient des variables d'environnements, de serveurs ou définies par l'utilisateur dans la portée d'appel de la fonction `get_defined_vars` .

8.135.12 `get_resource_type()` : Retourne le type de ressource

string `get_resource_type` (resource handle)

Cette fonction retourne le type de la ressource donnée.

8.135.13 `gettype()` : Retourne le type de la variable

string `gettype` (mixed var)

`gettype` retourne le type de la variable PHP var .

Attention

N'utilisez jamais `gettype` pour vérifier un type, car la chaîne retournée est sujette à modification sans préavis dans une version ultérieure. De plus, c'est bien plus long comme cela, car cela implique une comparaison de chaîne.

A la place, utilisez les fonctions `is_*` .

Les chaînes de caractères que peut retourner la fonction sont les suivantes :

- " boolean " (depuis PHP 4)
- " entier "
- " double " (pour des raisons historiques, " double " est retournée lorsqu'une valeur de type float est fournie, au lieu de la chaîne " float ")
- " string "
- " array "
- " object "
- " resource " (depuis PHP 4)
- " NULL " (depuis PHP 4)
- " user function " (PHP 3 uniquement, abandonné)
- " unknown type "

En PHP 4, il est recommandé d'utiliser `function_exists` et `method_exists` pour remplacer l'utilisation de `gettype` sur une fonction.

Voir aussi `settype` , `is_array` , `is_bool` , `is_float` , `is_int` , `is_null` , `is_numeric` , `is_object` , `is_resource` , `is_scalar` et `is_string` .

8.135.14 `import_request_variables()` : Importe les variables de GET/POST/Cookie dans l'environnement global

bool `import_request_variables` (string types , string prefix)

`import_request_variables` importe les variables GET/POST/Cookie dans l'environnement global. C'est pratique si vous avez désactivé `register_globals`, mais que vous voulez enregistrer des variables dans l'environnement global.

En utilisant le paramètre `types`, vous pouvez spécifier les variables que vous voulez importer. Vous pouvez utiliser 'G', 'P' et 'C' pour désigner respectivement GET, POST et Cookie. Ces caractères ne sont pas sensibles à la casse, et vous pouvez les combiner entre eux. POST inclut les fichiers téléchargés. Notez que l'ordre des lettres est important. En utilisant par exemple "GP", les variables de POST écraseront par les variables de GET, ayant le même nom. Toute lettre autre que G, P ou C, est ignorée.

Le paramètre `prefix` est utilisé comme un préfixe de nom de variable, qui sera ajoutée au début de tous les noms de variables importées. Si vous avez par exemple une variable GET du nom de "userid", et utilisez le préfixe "pref_", la variable ainsi générée s'appellera \$pref_userid.

Si vous souhaitez importer d'autres variables dans l'environnement global, comme \$_SERVER, considérez l'utilisation de la fonction `extract`.

Note

Bien que le paramètre `prefix` soit optionnel, il sera généré une alerte `E_NOTICE` si vous ne spécifiez pas de préfixe, ou si vous utilisez une chaîne vide comme préfixe. C'est potentiellement un trou de sécurité. Les alertes de ce niveau ne sont pas affichées si vous utilisez le niveau d'erreur par défaut.

Exemple avec `import_request_variables`

```
<?php
// Ce code va importer les variables GET et POST
// avec un préfixe "rvar_"
import_request_variables("gP", "rvar_");

echo $rvar_foo;
?>
```

Voir aussi `$_REQUEST`, `register_globals`, `variables pré-définies` et `extract`.

8.135.15 `intval()` : Retourne la valeur numérique entière équivalent d'une variable

`int intval (mixed var , int base)`

`intval` retourne la valeur numérique entière (entier) de la variable `var`, en convertissant la valeur dans la base spécifiée (par défaut en base 10).

8.135.16 `is_array()` : Détermine si une variable est un tableau

`bool is_array (mixed var)`

`is_array` détermine si la variable donnée est un tableau.

8.135.17 is_bool() : Détermine si une variable est un booléen

```
bool is_bool ( mixed var )
```

is_bool détermine si la variable donnée est un booléen.

8.135.18 is_callable() : Détermine si l'argument peut être appelé comme fonction

```
bool is_callable ( mixed var , bool syntax_only , string callable_name )
```

is_callable vérifie qu'une variable peut être appelée comme fonction. Cette fonction peut vérifier qu'une variable contient un nom de fonction valide, ou bien qu'elle contient un tableau, avec un objet et un nom de méthode.

Le paramètre `var` peut être le nom d'une fonction stockée dans une variable de chaîne, ou un objet, et un nom de méthode pour cet objet, sous la forme : `array($unObjet, 'NomDeMethode')`

Si l'argument `syntax_only` vaut `TRUE` , la fonction ne va vérifier que si `var` peut être une fonction ou une méthode. Il va simplement rejeter les variables qui ne sont pas des chaînes, ou des tableaux qui n'ont pas la bonne structure pour être utilisé comme fonction de callback. Les tableaux valides sont supposés n'avoir que deux entrées, le premier étant un objet ou une chaîne, et de second une chaîne.

L'argument `callable_name` reçoit le nom à utiliser. Dans l'exemple ci-dessous, il vaut "uneClasse:uneMethode" . Notez que, bien que `uneClasse:uneMethode()` puisse être appelé sous forme statique, ce n'est pas le cas.

Exemple avec is_callable

```
<?php
// Comment vérifier qu'une variable peut être appelée
// comme fonction ?

//
// Variable simple contenant une fonction
//

function uneFonction() {
}

$fonctionVariable = 'uneFonction';

var_dump( is_callable( $fonctionVariable, false, $callable_name )); // bool(true)

echo $callable_name, "\n"; // uneFonction

//
// Tableau contenant une méthode
//

class uneClasse {

    function uneMethode() {
    }

}

$unObjet = new uneClasse();
```

```
$methodeVariable = array($unObjet, 'uneMethode' );
var_dump(is_callable($methodeVariable, true, $callable_name )); // bool(true)
echo $callable_name, "\n"; // uneClasse:uneMethode
?>
```

Voir aussi [function_exists](#) et [method_exists](#) .

8.135.19 [is_double\(\)](#) : Alias de [is_float](#)

Cette fonction est un alias de : [is_float](#) .

8.135.20 [is_float\(\)](#) : Détermine si une variable est de type nombre décimal

bool [is_float](#) (mixed var)

Détermine si la variable donnée est de type nombre décimal.

Note

Pour tester si une variable est un nombre ou une chaîne numérique (comme les entrées de formulaire, qui sont toujours des chaînes), vous devez utiliser la fonction [is_numeric](#) .

8.135.21 [is_int\(\)](#) : Détermine si une variable est de type nombre entier

bool [is_int](#) (mixed var)

Détermine si la variable donnée est de type nombre entier.

Note

Pour tester si une variable est un nombre ou une chaîne numérique (comme les entrées de formulaire, qui sont toujours des chaînes), vous devez utiliser la fonction [is_numeric](#) .

8.135.22 [is_integer\(\)](#) : Alias de [is_int](#)

Cette fonction est un alias de : [is_int](#) .

8.135.23 [is_long\(\)](#) : Alias de [is_int](#)

Cette fonction est un alias de : [is_int](#) .

8.135.24 [is_null\(\)](#) : Indique si une variable vaut NULL

bool [is_null](#) (mixed var)

Indique si la variable donnée vaut NULL .

8.135.25 **is_numeric()** : Détermine si une variable est un type numérique

bool **is_numeric** (mixed var)

Détermine si la variable donnée est de type numérique. Les chaînes numériques sont composées optionnellement de signes, de n'importe quel nombre de chiffres, optionnellement d'une partie décimale ainsi qu'une partie exponentielle. +0123.45e6 est une valeur numérique valide. La notation hexadécimale (0xFF) est autorisée également, mais uniquement sans un signe, sans décimale et sans partie exponentielle.

8.135.26 **is_object()** : Détermine si une variable est de type objet

bool **is_object** (mixed var)

Détermine si la variable donnée est de type objet.

8.135.27 **is_real()** : Alias de **is_float**

Cette fonction est un alias de : [is_float](#) .

8.135.28 **is_resource()** : Détermine si une variable est une ressource

bool **is_resource** (mixed var)

Détermine si une variable est une ressource.

8.135.29 **is_scalar()** : Indique si une variable est un scalaire

bool **is_scalar** (mixed var)

Indique si la variable donnée est un scalaire.

Les variables scalaires sont celles qui contiennent des entiers, des nombres à virgule flottante, des chaînes de caractères ou des booléens. Les types array , object et resource ne sont pas scalaires.

Note

<p>is_scalar ne considère pas les valeurs des types ressource comme scalaires, étant donné que les ressources sont des types abstraits, basés sur des entiers. Ceci est susceptible de changer.</p>

8.135.30 **is_string()** : Détermine si une variable est de type chaîne de caractères

bool **is_string** (mixed var)

Détermine si la variable donnée est de type chaîne de caractères.

8.135.31 isset() : Détermine si une variable est affectée

bool **isset** (mixed var , *mixed* var , ...)

isset renvoie TRUE si la variable var est définie, FALSE sinon.

Si une variable a été détruite avec la fonction unset , la fonction isset renverra FALSE . isset renverra FALSE lors du test d'une variable de valeur NULL . Notez aussi que le caractère nul ("\0") n'est pas équivalent à la constante PHP NULL .

Note

Attention !

isset fonctionne uniquement avec des variables car l'utilisation de toute autre chose aura comme conséquence une erreur d'analyse. Pour vérifier si une constants est définie, utilisez la fonction defined .

Exemple avec isset

```

<?php

$var = '';

// Ceci est vrai, alors le texte est affiché
if (isset($var)) {
    echo 'Cette variable existe, donc je peux l\'afficher.';
}

// Dans les exemples suivants, nous utilisons var_dump() pour afficher
// le retour de la fonction isset().

$a = 'test';
$b = 'anothertest';

var_dump(isset($a));      // TRUE
var_dump(isset($a, $b)); // TRUE

unset ($a);

var_dump(isset($a));      // FALSE
var_dump(isset($a, $b)); // FALSE

$foo = NULL;
var_dump(isset($foo));    // FALSE

?>

```

Cela fonctionne aussi avec les tableaux :

Exemple avec isset et un tableau

```

<?php

$a = array ('test' => 1, 'bonjour' => NULL);

var_dump(isset($a['test']));      // TRUE
var_dump(isset($a['foo']));       // FALSE
var_dump(isset($a['bonjour']));   // FALSE

// La clé 'bonjour' vaut NULL est est considérée comme non existante
// Si vous voulez vérifier l'existence de cette clé, utilisez cette fonction
var_dump(array_key_exists('bonjour', $a) ); // TRUE

```

```
?>
```

Note

Comme ceci est une structure du langage, et non pas une fonction, il n'est pas possible de l'appeler avec les variables fonctions.

Voir aussi empty, unset, defined, le tableau de comparaison de types, array_key_exists et l'opérateur de contrôle de rapport d'erreur @.

8.135.32 print_r() : Affiche des informations lisibles pour une variable

bool **print_r** (mixed *expression* , bool *return*)

print_r affiche des informations à propos d'une variable, de manière à ce qu'elle soit lisible. Pour une chaîne, un entier ou un double, la valeur elle même sera affichée. Pour les tableaux, les valeurs seront présentées dans un format qui montre les clés et les valeurs. Une notation similaire est disponible pour les objets. print_r et var_export afficheront également les propriétés protégées et privées d'un objet en PHP 5, contrairement à var_dump.

Gardez en tête que print_r place le pointeur de tableau à la fin du tableau. Utilisez reset pour le ramener au début.

Exemple avec print_r

```
<pre>
<?php
$a = array ('a' => 'pomme',
           'b' => 'banane',
           'c' => array ('x',
                       'y',
                       'z'));

print_r ($a);
?>
</pre>
```

L'exemple ci-dessus va afficher :

```
<pre>
Array
(
[a] => pomme
[b] => banane
[c] => Array
(
[0] => x
[1] => y
[2] => z
)
)
</pre>
```

Si vous voulez obtenir le résultat de print_r dans une chaîne, utilisez le paramètre *return*. Si ce paramètre vaut TRUE, print_r retournera ce qu'il aurait affiché (ce qu'il fait par défaut).

Utilisation du paramètre *return* de print_r

```
<?php
$b = array ('m' => 'mangue', 'foo' => 'bar', 'x' => array ('x', 'y', 'z'));
$resultat = print_r ($b, true);
//$resultat contient maintenant le résultat de print_r()
?>
```

Note

Si vous devez récupérer le résultat de `print_r` avec une version de PHP plus ancienne que 4.3.0, utilisez les fonctions de [bufferisation de sortie](#) .

Note

Le paramètre `return` a été ajouté en PHP 4.3.0.

Note

Avant PHP 4.0.4, `print_r` bouclera à l'infini si un [tableau](#) ou un [objet](#) contient une référence sur lui-même. Un exemple classique est notamment la ligne `print_r($GLOBALS)` car `$GLOBALS` est lui-même une variable globale, qui donc, contient une référence sur elle même.

Voir aussi [ob_start](#) , [var_dump](#) et [var_export](#) .

8.135.33 `serialize()` : Linéarise une variable

string **serialize** (mixed value)

[serialize](#) retourne une chaîne contenant une représentation linéaire de valeur , pour stockage.

C'est une technique pratique pour stocker ou passer des valeurs de PHP entre scripts, sans perdre ni leur structure, ni leur type.

Pour récupérer une variable linéarisée, et retrouver une variable, utilisez [unserialize](#) . [serialize](#) acceptent tous les types sauf les ressources. Vous pouvez linéariser un tableau qui contient une référence sur lui-même. Les références dans ce tableau/objet seront également stockées.

Lors de la linéarisation d'un objet, PHP tentera d'appeler la fonction membre `__sleep` avant de linéariser. Cela permet à l'objet de faire un dernier nettoyage, etc. avant d'être linéarisé. De même, lorsque l'objet est restauré en utilisant [unserialize](#) , la fonction membre `__wakeup` est appelée.

Note

Ceci ne fonctionne pas correctement avant la version 4.0.7.

Note

En PHP 3, les propriétés de l'objet seront linéarisées, mais les méthodes seront perdues. Cette limitation a été enlevée en PHP 4 et restaure les propriétés et les méthodes. Merci de voir la section sur la [linéarisations d'objets](#) des [classes et objets](#) pour plus d'informations.

Il n'est pas possible de linéariser les objets internes de PHP.

Exemple avec [serialize](#)

```
<?php
// $session_data contient un tableau multi-dimensionnel , avec les
// informations de session de l'utilisateur courant. On utilise serialize()
// pour les stocker dans une base de données

$conn = odbc_connect("webdb", "php", "chicken");
$stmt = odbc_prepare($conn,
    "UPDATE sessions SET data = ? WHERE id = ?");
$sqldata = array(serialize($session_data), $PHP_AUTH_USER);
```

```

if (!odbc_execute($stmt, &$sqldata)) {
    $stmt = odbc_prepare($conn,
        "INSERT INTO sessions (id, data) VALUES(?, ?)");
    if (!odbc_execute($stmt, &$sqldata)) {
        /* Un problème est survenu ! */
    }
}
?>

```

Voir aussi [unserialize](#) .

8.135.34 settype() : Affecte un type à une variable

bool **settype** (mixed var , string type)

[settype](#) force le type de la variable var en type .

Les valeurs possibles pour le paramètre type sont :

- " boolean " (ou, depuis PHP 4.2.0, " bool ")
- " integer " (ou, depuis PHP 4.2.0, " int ")
- " float " (uniquement depuis PHP 4.2.0. Pour les anciennes versions, utilisez l'alternative " double ")
- " string "
- " array "
- " object "
- " null " (depuis PHP 4.2.0)

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec [settype](#)

```

$foo = "5bar"; // chaîne
$bar = true;   // booléen

settype($foo, "integer"); // $foo vaut maintenant 5 (integer)
settype($bar, "string");  // $bar vaut maintenant "1" (string)

```

Voir aussi [gettype](#) , [transtypage](#) et [définition du type](#) .

8.135.35 strval() : Retourne la valeur de la variable, au format chaîne

string **strval** (mixed var)

[strval](#) retourne la valeur de la variable var , au format chaîne de caractères. Voir la documentation sur les chaînes de caractères pour plus d'informations sur la conversion en chaîne.

var peut être un scalaire. Vous ne pouvez pas utiliser la fonction [strval](#) avec des tableaux ou des objets.

Voir aussi [doubleval](#) , [intval](#) , [settype](#) et [Transtypage](#) .

8.135.36 unserialize() : Crée une variable PHP à partir d'une valeur linéarisée

mixed **unserialize** (string str)

unserialize prend une variable linéarisée (voir serialize) et la convertit en variable PHP. La valeur convertie est retournée par la fonction, et peut être de type boolean , entier , float , string , array ou object . Si la chaîne passée ne peut être délinéarisée, cette fonction retourne FALSE et une erreur E_NOTICE est émise.

Attention

FALSE est retourné dans les cas où il y a une erreur et dans la délinéarisation de la valeur linéarisée FALSE . Le cas spécial peut être analysé en comparant str avec serialize(false) ou en attrapant l'erreur E_NOTICE émise.

Note

Directive unserialize_callback_func

Il est possible de définir une fonction de callback qui sera appelée si une classe indéfinie est utilisée lors de la délinéarisation (ce qui évitera de voir l'objet recevoir le type d'objet incomplet object " __PHP_Incomplete_Class "). Utilisez dans votre fichier php.ini ou le fichier .htaccess , ou encore avec la fonction ini_set , pour définir une fonction unserialize_callback_func() . Chaque fois qu'une classe non-définie sera instanciée, cette fonction sera appelée. Pour désactiver cette fonctionnalité, laissez-la simplement vide. Notez aussi que cette directive n'est disponible que depuis PHP 4.2.0.

Si la variable délinéarisée est un objet, après avoir réussi à le reconstruire, PHP appellera automatiquement la méthode __wakeup() si elle existe.

Exemple avec la directive unserialize_callback_func

```
<?php
$serialized_object='O:1:"a":1:{s:5:"value";s:3:"100";}';

// directive unserialize_callback_func disponible depuis PHP 4.2.0
ini_set('unserialize_callback_func','moncallback');

// configure votre fonction de callback : moncallback

function moncallback($classname)
{
    // Incluez simplement in fichier contenant votre définition de classe
    // vous saurez quelle classe grâce à $classname
}
?>
```

Note

En PHP 3, les méthodes ne sont pas préservées lors de la délinéarisation d'un objet. Cette limitation a été levée en PHP 4 et stocke désormais les méthodes et les membres. Voyez le chapitre sur la Linéarisation des objets du chapitre Classes et Objets .

Exemple avec unserialize

```
<?php
// Ici, on utilise <function>unserialize</function> pour charger les données de sessions
// depuis la base de données, dans $session_data. Cet exemple complète
// celui fourni avec <function>serialize</function>.
$conn = odbc_connect("webdb", "php", "chicken");
$stmt = odbc_prepare($conn, "SELECT data FROM sessions WHERE id = ?");
$sqldata = array($PHP_AUTH_USER);
```

```

if (!odbc_execute($stmt, &$sqldata) || !odbc_fetch_into($stmt, &$tmp)) {
    // si la préparation ou la lecture échoue, on crée un tableau vide
    $session_data = array();
} else {
    // les données sauvées sont dans $tmp[0].
    $session_data = unserialize($tmp[0]);
    if (!is_array($session_data)) {
        // Erreur... initialisation à tableau vide
        $session_data = array();
    }
}
?>

```

Voir aussi [serialize](#) .

8.135.37 unset() : Détruit une variable

void **unset** (mixed var , *mixed var* , *mixed ...*)

unset détruit les variables spécifiées. Notez qu'en PHP 3, unset retournait toujours TRUE (en fait, la valeur entière 1). En PHP 4 toutefois, unset n'est plus une véritable fonction : c'est une structure du langage, ce qui fait qu'elle ne retourne pas de valeur. Lire la valeur retournée par unset (dans une variable, par exemple), retourne une erreur d'analyse.

Exemple avec unset

```

<?php
// Destruction d'une seule variable
unset($foo);
// Destruction d'un élément de tableau
unset($bar['quux']);
// Destruction de plusieurs variables
unset($foo1, $foo2, $foo3);
?>

```

Le comportement de unset à l'intérieur d'une fonction peut varier suivant le type de variable que vous voulez détruire.

Si une variable globale est détruite avec unset depuis une fonction, seule la variable locale sera détruite. Le variable globale gardera la valeur acquise avant l'appel à unset .

Exemple avec unset dans un contexte de fonction

```

<?php
function destroy_foo()
{
    global $foo;
    unset($foo);
}
$foo = 'bar';
destroy_foo();
echo $foo;
?>

```

L'exemple ci-dessus va afficher :

bar

Si vous désirez détruire une variable globale à l'intérieur d'une fonction, vous pouvez utiliser le tableau \$GLOBALS pour le faire :

Exemple avec `unset` avec une variable globale

```
<?php
function foo()
{
    unset($GLOBALS['bar']);
}

$bar = "truc";
foo();
?>
```

Si une variable qui est passée par référence est détruite à l'intérieur d'une fonction, seule la variable locale sera détruite. La variable globale conservera la même valeur qu'elle avait avant l'appel de `unset`.

Exemple avec `unset` sur une référence

```
<?php
function foo(&$bar)
{
    unset($bar);
    $bar = "bla";
}

$bar = 'truc';
echo "$bar\n";
foo($bar);
echo "$bar\n";
?>
```

L'exemple ci-dessus va afficher :

```
truc
truc
```

Si une variable statique est détruite à l'intérieure d'une fonction `unset` détruira la variable seulement dans le contexte du reste de la fonction. Les appels suivants restaureront la valeur précédente de la variable.

Exemple avec `unset` et une variable statique

```
<?php
function foo()
{
    static $bar;
    $bar++;
    echo "Avant unset : $bar, ";
    unset($bar);
    $bar = 23;
    echo "après unset : $bar\n";
}

foo();
foo();
foo();
?>
```

L'exemple ci-dessus va afficher :

Avant unset : 1, après unset : 23
Avant unset : 2, après unset : 23
Avant unset : 3, après unset : 23

Note

Comme ceci est une structure du langage, et non pas une fonction, il n'est pas possible de l'appeler avec les variables fonctions .

Voir aussi array_splice , isset et empty .

8.135.38 **var_dump()** : Affiche les informations d'une variable

void **var_dump** (mixed expression , *mixed expression* , ...)

var_dump retourne les informations structurées d'une variable, y compris son type et sa valeur. Les tableaux et les objets sont explorés récursivement, avec des indentations, pour mettre en valeur leur structure.

En PHP 5, seules les propriétés publiques, privées et protégées seront retournées dans le résultat.

Array

8.135.39 **var_export()** : Retourne le code PHP utilisé pour générer une variable

mixed **var_export** (mixed expression , *bool return*)

var_export retourne des données structurées sur la variable donnée. C'est le même principe que var_dump mais avec une exception : le résultat retourné est du code PHP valide.

8.136 vpopmail

8.136.1 Introduction

Attention

Ce module est **EXPERIMENTAL** . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez ce module à vos risques et périls.

Cette extension a été déplacée vers PECL depuis PHP 4.3.0.

8.136.2 Installation

En PHP 4, ces fonctions ne sont disponibles que si PHP a été configurée avec l'option `--with-vpopmail[=DIR]` .

- [Introduction](#)
- [Installation](#)
- [vpopmail_add_alias_domain_ex](#)
- [vpopmail_add_alias_domain](#)
- [vpopmail_add_domain_ex](#)
- [vpopmail_add_domain](#)
- [vpopmail_add_user](#)
- [vpopmail_alias_add](#)
- [vpopmail_alias_del_domain](#)
- [vpopmail_alias_del](#)
- [vpopmail_alias_get_all](#)
- [vpopmail_alias_get](#)
- [vpopmail_auth_user](#)
- [vpopmail_del_domain_ex](#)
- [vpopmail_del_domain](#)
- [vpopmail_del_user](#)
- [vpopmail_error](#)
- [vpopmail_passwd](#)
- [vpopmail_set_user_quota](#)

8.136.4 vpopmail_add_alias_domain() : Ajout un alias pour un domaine virtuel

bool **vpopmail_add_alias_domain** (string domain , string aliasdomain)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.136.5 vpopmail_add_domain_ex() : Ajoute un nouveau domaine virtuel

bool **vpopmail_add_domain_ex** (string domain , string passwd , *string* quota , *string* bounce , *bool* apop)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.136.6 vpopmail_add_domain() : Ajoute un nouveau domaine virtuel

bool **vpopmail_add_domain** (string domain , string dir , int uid , int gid)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.136.7 vpopmail_add_user() : Ajoute un nouvel utilisateur à un domaine virtuel

bool **vpopmail_add_user** (string user , string domain , string password , *string* gecos , *bool* apop)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.136.8 vpopmail_alias_add() : Ajoute un alias virtuel

bool **vpopmail_alias_add** (string user , string domain , string alias)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.136.9 vpopmail_alias_del_domain() : Efface tous les alias virtuels d'un domaine

bool **vpopmail_alias_del_domain** (string domain)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.136.10 vpopmail_alias_del() : Efface tous les alias d'un utilisateur

bool **vpopmail_alias_del** (string user , string domain)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.136.11 vpopmail_alias_get_all() : Lit toutes les lignes d'alias d'un domaine

array **vpopmail_alias_get_all** (string domain)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.136.12 vpopmail_alias_get() : Lit toutes les lignes d'un alias de domaine

array **vpopmail_alias_get** (string alias , string domain)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.136.13 vpopmail_auth_user() : Valide le mot de passe d'un utilisateur pour un domaine

bool **vpopmail_auth_user** (string user , string domain , string password , string apop)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.136.14 vpopmail_del_domain_ex() : Efface un domaine virtuel

bool **vpopmail_del_domain_ex** (string domain)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.136.15 vpopmail_del_domain() : Efface un domaine virtuel

bool **vpopmail_del_domain** (string domain)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.136.16 vpopmail_del_user() : Efface un utilisateur d'un domaine virtuel

bool **vpopmail_del_user** (string user , string domain)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.136.17 vpopmail_error() : Lit le dernier message d'erreur vpopmail

string **vpopmail_error** (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.136.18 vpopmail_passwd() : Remplace le mot de passe d'un utilisateur virtuel

bool **vpopmail_passwd** (string user , string domain , string password , bool *apop*)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.136.19 vpopmail_set_user_quota() : Modifie le quota d'un utilisateur virtuel

bool **vpopmail_set_user_quota** (string user , string domain , string quota)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.137 API windows

8.137.1 Introduction

Cette extension est une API générique d'accès aux DLL. Elle a été originalement écrite pour accéder aux API Win32 depuis PHP, mais vous pouvez l'utiliser pour accéder à d'autres fonctions exportées par des DLL.

Actuellement, les types supportés sont les types PHP génériques (chaînes de caractères, booléens, nombres, entiers et NULL), et les types que vous définissez avec `w32api_deftype`.

Note
Cette extension a été déplacée dans le module <code>PECL</code> et ne sera plus intégrée dans PHP à partir de PHP 5.1.0.
Attention
Ce module est EXPERIMENTAL . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez ce module à vos risques et périls.

8.137.2 Pré-requis

Cette extension ne fonctionne qu'avec le système Windows.

8.137.3 Installation

Il n'y a pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

8.137.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.137.5 Types de ressources

Cette extension définit un type de ressource, utilisé pour les types de données créé par l'utilisateur. Le nom de cette ressource est "dynaparm".

8.137.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

DC_MICROSOFT ([entier](#))
 DC_BORLAND ([entier](#))
 DC_CALL_CDECL ([entier](#))
 DC_CALL_STD ([entier](#))
 DC_RETVAL_MATH4 ([entier](#))

DC_RETVAl_MATH8 ([entier](#))
 DC_CALL_STD_BO ([entier](#))
 DC_CALL_STD_MS ([entier](#))
 DC_CALL_STD_M8 ([entier](#))
 DC_FLAG_ARGPTR ([entier](#))

8.137.7 Exemples

Cet exemple lit la durée de vie du système, et l'affiche dans une boîte :

Lit la durée de vie de la machine, et l'affiche
<pre> <?php // Définit les constantes nécessaire, issues de // Visual Studio/Tools/Winapi/WIN32API.txt define("MB_OK", 0); // Chargement de l'extension dl("php_w32api.dll"); // Enregistrement de la fonction GetTickCount, provenant de kernel32.dll w32api_register_function("kernel32.dll", "GetTickCount", "long"); // Enregistrement de la fonction MessageBoxA issue de User32.dll w32api_register_function("User32.dll", "MessageBoxA", "long"); // Lecture de la durée de vie de la machine \$ticks = GetTickCount(); // Conversion en texte lisible \$secs = floor(\$ticks / 1000); \$mins = floor(\$secs / 60); \$hours = floor(\$mins / 60); \$str = sprintf("Vous utilisez cet ordinateur depuis : ". "\r\n %d millisecondes, ou \r\n %d secondes". "ou \r\n %d minutes ou\r\n %d heures %d minutes.", \$ticks, \$secs, \$mins, \$hours, \$mins - (\$hours*60)); // Affiche le message dans une boîte de dialogue avec un bouton OK MessageBoxA(NULL, \$str, "Uptime Information", MB_OK); ?> </pre>

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)

- [Exemples](#)
- [w32api_deftype](#)
- [w32api_init_dtype](#)
- [w32api_invoke_function](#)
- [w32api_register_function](#)
- [w32api_set_call_method](#)

8.137.9 w32api_init_dtype() : Crée une instance de type de données Win32 et la remplit

resource **w32api_init_dtype** (string typename , mixed value , *mixed* ...)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[w32api_init_dtype](#) crée une instance du type typename , et remplit les valeurs avec les données value , Le nom du type typename est sensible à la casse. Il faut donner les valeurs dans le même ordre que lors de la définition du type, avec [w32api_deftype](#) . Le type de la ressource retournée est dynaparm .

8.137.10 w32api_invoke_function() : Appelle une fonction windows 32

mixed **w32api_invoke_function** (string funcname , mixed argument , *mixed* ...)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[w32api_invoke_function](#) essaie de trouver la fonction préalablement enregistrée, appelée funcname , en lui passant les arguments argument , ... , passés après funcname . Le type retourné est celui que vous avez enregistré avec la fonction. La valeur est celle retournée par la fonction. Les arguments peuvent être n'importe lequel de ceux de PHP, ou un type défini via la fonction [w32api_deftype](#) .

8.137.11 w32api_register_function() : Enregistre une fonction Win32 dans PHP

bool **w32api_register_function** (string library , string function_name , string return_type)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

[w32api_register_function](#) essaie de trouver la fonction function_name dans la bibliothèque library , et essaie de l'importer en PHP. La fonction sera alors enregistrée avec le type de valeur de retour return_type . return_type peut être un type générique de PHP ou bien un type défini par le programmeur avec la fonction [w32api_deftype](#) . Tous les noms de type sont sensibles à la casse. Les types intégrés doivent être fournis en minuscules. Cette fonction retourne TRUE en cas de

succès, FALSE en cas d'échec.

8.137.12 `w32api_set_call_method()` : Modifie le type d'appel de la fonction Win32

void `w32api_set_call_method` (int method)

Attention

Cette fonction est EXPERIMENTALE . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.
--

`w32api_set_call_method` modifie le type d'appel de la fonction. Le paramètre `method` peut être l'une des constantes suivantes : `DC_CALL_CDECL` ou `DC_CALL_STD` . Par défaut, c'est `DC_CALL_STD` .

8.138 WDDX

8.138.1 Introduction

Ces fonctions fonctionnent grâce à [WDDX](#) .

8.138.2 Pré-requis

Pour utiliser WDDX, vous devez installer la bibliothèque EXPAT (qui est fournie avec la distribution d'Apache 1.3.7 ou plus récent).

8.138.3 Installation

Après avoir installé Expat, compilez PHP avec `--enable-wddx` .

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de bibliothèque supplémentaire pour disposer de ces fonctions.

8.138.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.138.5 Types de ressources

Cette extension ne définit aucune ressource.

8.138.6 Constantes pré-définies

Cette extension ne définit aucune constante.

8.138.7 Exemples

Notez bien que toutes les fonctions qui enregistrent des données, utilisent le premier élément d'un tableau pour savoir si ce tableau doit être enregistré sous la forme d'un tableau, ou d'une structure. Si le premier élément a une clé de type chaîne, le tableau sera enregistré sous la forme d'une structure, et sinon, sous la forme d'un tableau.

Enregistre une valeur simple

```
<?php
echo wddx_serialize_value("Exemple de paquet de PHP à WDDX", "Paquet PHP");
?>
```

Cet exemple va produire le résultat suivant :

```
<wddxPacket version='1.0'><header><comment>Paquet PHP</comment></header><data><string>Exemple de
```

Utilisation de paquets incrémentaux

```

<?php
$pi = 3.1415926;
$packet_id = wddx_packet_start("PHP");
wddx_add_vars($packet_id, "pi");

/* Supposons que $villes provient d'une base de données */
$villes = array("Paris", "Montréal", "Lyon");
wddx_add_vars($packet_id, "villes");

$packet = wddx_packet_end($packet_id);
echo $packet;
?>

```

Cet exemple donnera :

```

<wddxPacket version='1.0'><header><comment>PHP</comment></header><data><struct><var
name='pi'><number>3.1415926</number></var><var name='villes'><array
length='3'><string>Paris</string><string>Montréal</string><string>Lyon</string></array></var></struct></d

```

Note

Si vous voulez linéariser des caractères non-ASCII, vous devez d'abord convertir vos données en UTF-8. (voir les fonctions [utf8_encode](#) et [iconv](#)).

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [wddx_add_vars](#)
- [wddx_deserialize](#)
- [wddx_packet_end](#)
- [wddx_packet_start](#)
- [wddx_serialize_value](#)
- [wddx_serialize_vars](#)
- [wddx_unserialize](#)

8.138.9 wddx_deserialize() : Alias de [wddx_unserialize](#)

Cette fonction est un alias de : [wddx_unserialize](#) .

8.138.10 wddx_packet_end() : Clôt un paquet WDDX

string [wddx_packet_end](#) (resource packet_id)

[wddx_packet_end](#) clôt un paquet WDDX repéré par son identifiant packet_id .

8.138.11 `wddx_packet_start()` : Commence un nouveau paquet WDDX avec une structure

resource `wddx_packet_start` (*string comment*)

`wddx_packet_start` sert à créer un nouveau paquet WDDX, pour pouvoir y faire des ajouts incrémentaux de variables. Cette fonction prend un argument optionnel `comment` et retourne un identifiant de paquet, qui servira à d'autres fonctions. Elle va automatiquement créer une définition de structure dans le paquet, pour accueillir des variables.

8.138.12 `wddx_serialize_value()` : Enregistre une valeur dans un paquet WDDX

string `wddx_serialize_value` (mixed *var* , *string comment*)

`wddx_serialize_value` sert à créer un paquet WDDX à partir d'une seule valeur. Cette fonction prend la valeur de `var` , et un argument optionnel `comment` qui apparaîtra dans l'en-tête du paquet, et retourne un paquet WDDX.

8.138.13 `wddx_serialize_vars()` : Enregistre plusieurs valeurs dans un paquet WDDX

string `wddx_serialize_vars` (mixed *var_name* , *mixed ...*)

`wddx_serialize_vars` sert à créer un paquet WDDX avec une structure qui contient la représentation des variables passées en arguments.

`wddx_serialize_vars` prend un nombre variable d'arguments, chacun d'entre eux pouvant être une chaîne contenant le nom d'une variable, ou un tableau de chaîne de nom de variable, ou même d'autres tableaux.

<code>wddx_serialize_vars</code>
<pre><?php \$a = 1; \$b = 5.5; \$c = array("bleu", "orange", "violet"); \$d = "colors"; \$clvars = array("c", "d"); echo wddx_serialize_vars("a", "b", \$clvars); ?></pre>

L'exemple ci-dessus donnera :

```
<wddxPacket version='1.0'><header/><data><struct><var name='a'><number>1</number></var>
<var name='b'><number>5.5</number></var><var name='c'><array length='3'>
<string>bleu</string><string>orange</string><string>violet</string></array></var>
<var name='d'><string>colors</string></var></struct></data></wddxPacket>
```

8.138.14 `wddx_unserialize()` : Délinéarise un paquet WDDX

mixed `wddx_unserialize` (string packet)

`wddx_unserialize` prend une chaîne packet et la délinéarise. Elle retourne le résultat qui peut être une chaîne, un nombre ou un tableau. Notez que les structures sont délinéarisées dans des tableaux associatifs.

8.139 Fonctions xattr

8.139.1 Introduction

L'extension xattr vous permet de manipuler les attributs étendus sur un système de fichiers.

8.139.2 Pré-requis

Pour utiliser xattr, vous avez besoin de la bibliothèque libattr . Elle est disponible sur <http://oss.sgi.com/projects/xfs/> .

Note

Ces fonctions fonctionnent uniquement sur les systèmes de fichiers qui supportent les attributs étendus et qui les ont activés lors du montage. Les systèmes de fichiers les plus courant supportant les attributs étendus sont ext2 , ext3 , reiserfs , jfs et xfs .

8.139.3 Installation

xattr est actuellement disponible via PECL <http://pecl.php.net/package/xattr> .

Si [PEAR](#) est disponible sur votre système *nix-like, vous pouvez utiliser l'installateur Pear pour installer l'extension xattr, via la commande : `pear -v install xattr` .

Vous pouvez toujours télécharger le paquet tar.gz et installer manuellement xattr :

Installation manuelle de l'extension xattr

```
gunzip xattr-xxx.tgz
tar -xvf xattr-xxx.tar
cd xattr-xxx
phpize
./configure && make && make install
```

8.139.4 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

XATTR_ROOT (entier)

Définit un attribut dans l'espace de noms racine. Nécessaire les privilèges d'administrateur.

XATTR_DONTFOLLOW (entier)

Ne pas suivre un lien symbolique mais opérer directement sur le lien symbolique.

XATTR_CREATE (entier)

La fonction échouera si l'attribut étendu existe déjà.

XATTR_REPLACE (entier)

La fonction échouera si l'attribut étendu n'existe pas.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Constantes pré-définies](#)

- [xattr_get](#)
- [xattr_list](#)
- [xattr_remove](#)
- [xattr_set](#)
- [xattr_supported](#)

8.139.6 [xattr_list\(\)](#) : Récupère une liste d'attributs étendus

array [xattr_list](#) (string filename , int flags)

[xattr_list](#) récupère une liste de noms d'attributs étendus d'un fichier désigné par le paramètre path .

Les attributs étendus ont deux espaces de noms différents : user et root . L'espace de noms user est disponible pour tous les utilisateurs tandis que l'espace de noms root n'est disponible que pour les utilisateurs ayant les privilèges root . [xattr](#) opère sur l'espace de noms user par défaut, mais vous pouvez changer cela en utilisant l'argument flags .

8.139.7 [xattr_remove\(\)](#) : Efface un attribut étendu

bool [xattr_remove](#) (string filename , string name , int flags)

[xattr_remove](#) efface un attribut étendu nommé name d'un fichier désigné par le paramètre path .

Les attributs étendus ont deux espaces de noms différents : user et root . L'espace de noms user est disponible pour tous les utilisateurs tandis que l'espace de noms root n'est disponible que pour les utilisateurs ayant les privilèges root . [xattr](#) opère sur l'espace de noms user par défaut, mais vous pouvez changer cela en utilisant l'argument flags .

8.139.8 [xattr_set\(\)](#) : Définie un attribut étendu

bool [xattr_set](#) (string filename , string name , string value , int flags)

[xattr_set](#) définit la valeur d'un attribut étendu nommé name à la valeur value d'un fichier désigné par le paramètre path .

Les attributs étendus ont deux espaces de noms différents : user et root . L'espace de noms user est disponible pour tous les utilisateurs tandis que l'espace de noms root n'est disponible que pour les utilisateurs ayant les privilèges root . [xattr](#) opère sur l'espace de noms user par défaut, mais vous pouvez changer cela en utilisant l'argument flags .

8.139.9 [xattr_supported\(\)](#) : Vérifie si un système de fichier supporte les attributs étendus

bool [xattr_supported](#) (string filename , int flags)

[xattr_supported](#) vérifie si le système de fichier contenant le fichier path supporte les attributs étendus. Un accès en lecture au fichier path est nécessaire.

8.140 Fonctions xdiff

8.140.1 Introduction

L'extension xdiff crée et applique des patches aux fichiers binaires et textes.

8.140.2 Pré-requis

Pour utiliser xdiff, vous aurez besoin de libxdiff, disponible sur le site web de libxdiff <http://www.xmailserver.org/xdiff-lib.html> .

Note

Vous devez avoir au moins la version 0.7 de la bibliothèque libxdiff pour que ces fonctions prennent en compte le paramètre `memory_limit` .

8.140.3 Installation

xdiff est actuellement disponible dans PECL <http://pecl.php.net/package/xdiff> .

Si PEAR est disponible sur votre système *nix vous pouvez utiliser l'installateur PEAR pour installer l'extension xdiff avec la commande suivante : `pear -v install xdiff` .

Vous pouvez toujours télécharger le tar.gz et installer xdiff à la main :

Installation à la main de xdiff

```
gunzip xdiff-xxx.tgz
tar -xvf xdiff-xxx.tar
cd xdiff-xxx
phpize
./configure && make && make install
```

8.140.4 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

`XDIFF_PATCH_NORMAL` (entier)
`XDIFF_PATCH_REVERSE` (entier)

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Constantes pré-définies](#)
- [xdiff_file_diff_binary](#)
- [xdiff_file_diff](#)
- [xdiff_file_merge3](#)
- [xdiff_file_patch_binary](#)
- [xdiff_file_patch](#)
- [xdiff_string_diff_binary](#)
- [xdiff_string_diff](#)
- [xdiff_string_merge3](#)

- [xdiff_string_patch_binary](#)
- [xdiff_string_patch](#)

8.140.6 `xdiff_file_diff()` : Créé un diff unifié entre deux fichiers

`bool xdiff_file_diff (string file1 , string file2 , string dest , int context , bool minimal)`

`xdiff_file_diff` crée un diff unifié des fichiers `file1` et `file2` et stocke le résultat dans le fichier `dest` . `context` indique le nombre de lignes de contexte que vous voulez inclure dans le résultat de diff. Configurez `minimal` à `TRUE` si vous voulez minimaliser la taille du diff (peut prendre beaucoup de temps). Le fichier résultant est humainement lisible.

Note

Cette fonction ne fonctionne pas correctement avec les fichiers binaires. Pour effectuer un diff sur des fichiers binaires, utilisez la fonction [xdiff_file_diff_binary](#) .

Cette fonction retourne `TRUE` en cas de succès, `FALSE` en cas d'échec.

Exemple avec `xdiff_file_diff`

Le code suivant crée un diff de deux fichiers php.

```
<?php
$ancienne_version = 'mon_script.php';
$nouvelle_version = 'mon_nouveau_script.php';

xdiff_file_diff($ancienne_version, $nouvelle_version, 'mon_script.diff', 2);
?>
```

Voir aussi [xdiff_string_diff](#) .

8.140.7 `xdiff_file_merge3()` : Fusionne trois fichiers en un seul

`mixed xdiff_file_merge3 (string file1 , string file2 , string file3 , string dest)`

`xdiff_file_merge3` fusionne les fichiers `file1` , `file2` et `file3` en un seul et le stocke dans `dest` .

Cette fonction retourne `TRUE` si la fusion a réussi, une chaîne avec les parties rejetées s'il y'en a, ou `FALSE` si une erreur interne s'est produite.

Exemple avec `xdiff_file_merge3`

Le code suivant fusionne trois fichiers en un.

```
<?php
$ancienne_version = 'script_original.php';
$fix1 = 'script_avec_fix1.php';
$fix2 = 'script_avec_fix2.php';

$errors = xdiff_file_merge3($ancienne_version, $fix1, $fix2, 'fixed_script.php');
if (is_string($errors)) {
    echo "Rejets :\n";
    echo $errors;
}
?>
```

Voir aussi [xdiff_string_merge3](#) .

8.140.8 `xdiff_file_patch_binary()` : Patche un fichier avec un diff binaire

bool `xdiff_file_patch_binary` (string file , string patch , string dest)

`xdiff_file_patch_binary` patche le fichier file avec le patch binaire contenu dans le fichier patch et stocke le résultat dans dest .

Note

Les deux fichiers (le fichier et le patch) seront chargés en mémoire ; assurez-vous d'avoir défini `memory_limit` à une valeur assez élevée.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec `xdiff_file_patch_binary`

Le code suivant applique un diff binaire à un fichier.

```

<?php
$ancienne_version = 'archive-1.0.tgz';
$patch = 'archive.bpatch';

$result = xdiff_file_patch_binary($ancienne_version, $patch, 'archive-1.1.tgz');
if ($result) {
    echo "Fichier patché";
} else {
    echo "Le fichier n'a pu être patché";
}

?>

```

Voir aussi `xdiff_string_patch_binary` .

8.140.9 `xdiff_file_patch()` : Patche un fichier avec un diff unifié

mixed `xdiff_file_patch` (string file , string patch , string dest , int flags)

`xdiff_file_patch` patche le fichier file avec le patch unifié dans le fichier patch et stocke le résultat dans le fichier dest .

flags peut être soit `XDIFF_PATCH_NORMAL` (mode par défaut, patch normal) ou `XDIFF_PATCH_REVERSE` (patch inversé).

Cette fonction retourne FALSE si une erreur interne s'est produite, une chaîne avec les parties rejetées du patch ou TRUE si le patch a été appliqué avec succès.

Exemple avec `xdiff_file_patch`

Le code suivant applique un diff à un fichier.

```

<?php
$ancienne_version = 'mon_script-1.0.php';
$patch = 'mon_script.patch';

$errors = xdiff_file_patch($ancienne_version, $patch, 'my_script-1.1.php');
if (is_string($errors)) {
    echo "Rejets :\n";
    echo $errors;
}

```

?>

Exemple de retour à l'état initial après l'application d'un patch

Le code suivant annule un patch.

```
<?php
$nouvelle_version = 'mon_script-1.1.php';
$patch = 'mon_script.patch';

$errors = xdiff_file_patch($nouvelle_version, $patch, 'mon_script-1.0.php', XDIFF_PATCH_REVERSE);
if (is_string($errors)) {
    echo "Rejets :\n";
    echo $errors;
}
?>
```

Voir aussi [xdiff_string_patch](#) .

8.140.10 xdiff_string_diff_binary() : Créé un diff binaire de deux chaînes

string **xdiff_string_diff_binary** (string str1 , string str2)

[xdiff_string_diff_binary](#) crée un diff binaire des chaînes str1 et str2 .

Retourne une chaîne avec le résultat ou FALSE si une erreur interne est survenue.

Voir aussi [xdiff_file_diff_binary](#) .

8.140.11 xdiff_string_diff() : Créé un diff unifié entre deux chaînes

string **xdiff_string_diff** (string str1 , string str2 , int context , bool minimal)

[xdiff_string_diff](#) crée un diff unifié entre les deux chaînes str1 et str2 . context indique le nombre de lignes de contexte que vous voulez inclure dans le diff résultant. Configurez minimal à TRUE si vous voulez minimaliser la taille du diff (peut prendre beaucoup de temps).

Note

Cette fonction ne fonctionne pas correctement avec des chaînes binaires. Pour effectuer un diff de chaînes binaires, utilisez la fonction [xdiff_string_diff_binary](#) .

Retourne une chaîne avec le résultat ou FALSE si une erreur interne est survenue.

Exemple avec [xdiff_string_diff](#)

Le code suivant effectue un diff de deux articles.

```
<?php
$ancien_article = file_get_contents('./ancien_article.txt');
$nouvel_article = $_REQUEST['article']; /* disons que quelqu'un a posté un nouvel article via un

$diff = xdiff_string_diff($ancien_article, $nouvel_article, 1);
if (is_string($diff)) {
    echo "Différences entre les deux articles :\n";
    echo $diff;
}
```

```
?>
```

Voir aussi [xdiff_file_diff](#) .

8.140.12 `xdiff_string_merge3()` : Fusionne trois chaînes en une seule

```
mixed xdiff_string_merge3 ( string str1 , string str2 , string str3 , string error )
```

`xdiff_string_merge3` fusionne les chaînes `str1` , `str2` et `str3` en une seule.

Si `error` est fourni, alors les parties rejetées `y` sont stockées.

Retourne la chaîne fusionnée, `FALSE` si une erreur interne survient ou `TRUE` si la chaîne fusionnée est vide.

Voir aussi [xdiff_file_merge3](#) .

8.140.13 `xdiff_string_patch_binary()` : Patche une chaîne avec un diff binaire

```
string xdiff_string_patch_binary ( string str , string patch )
```

`xdiff_string_patch_binary` patche la chaîne `str` avec le patch binaire contenu dans la chaîne `patch` .

Retourne une chaîne patchée.

Voir aussi [xdiff_file_patch_binary](#) .

8.140.14 `xdiff_string_patch()` : Patche une chaîne avec un diff unifié

```
string xdiff_string_patch ( string str , string patch , int flags , string error )
```

`xdiff_string_patch` patche la chaîne `str` avec le patch unifié contenu dans la chaîne `patch` .

`flags` peut être soit `XDIFF_PATCH_NORMAL` (mode par défaut, patch normal), soit `XDIFF_PATCH_REVERSE` (patch inversé).

Si `error` est fourni, les parties rejetées `y` sont stockées.

Exemple avec `xdiff_string_patch`

Le code suivant applique les changements à des articles.

```
<?php
$aancien_article = file_get_contents('./ancien_article.txt');
$difff = $_SERVER['patch']; /* disons que quelqu'un a passé un patch via un formulaire html */
$errors = '';

$nouvel_article = xdiff_string_patch($ancien_article, $difff, XDIFF_PATCH_NORMAL, $errors);
if (is_string($nouvel_article)) {
    echo "Nouvel article:\n";
    echo $nouvel_article;
}
```

```
}  
if (strlen($errors)) {  
    echo "Rejets : \n";  
    echo $errors;  
}  
?>
```

Retourne une chaîne patchée.

Voir aussi [xdiff_file_patch](#) .

8.141 Analyseur syntaxique XML

8.141.1 Introduction

Le langage XML (eXtensible Markup Language (Langage à Balises Extensible)) est un format structuré de données pour les échanges sur le web. C'est un standard défini par le consortium World Wide Web (W3C). Plus d'informations à propos du XML et des technologies afférentes sont accessibles (en anglais) <http://www.w3.org/XML/> .

Cette extension PHP implémente la bibliothèque expat de James Clark sous PHP. Cela vous permettra d'analyser mais pas de valider les documents XML. Il supporte trois types de jeux de caractères différents, disponibles aussi sous PHP : US-ASCII , ISO-8859-1 et UTF-8 . UTF-16 n'est pas supporté.

Cette extension vous permet de créer des analyseurs XML puis de définir des **gestionnaires** pour chaque événement XML. Les analyseurs XML disposent de quelques paramètres que vous pouvez régler.

8.141.2 Pré-requis

Cette extension PHP utilise expat compat layer par défaut. Elle peut aussi utiliser expat , qui est disponible sur <http://www.jclark.com/xml/expat.html> . Le fichier Makefile livré avec expat ne construit pas par défaut de bibliothèque : il faut utiliser la ligne suivante :

```
libexpat.a: $(OBJS)
    ar -rc $@ $(OBJS)
    ranlib $@
```

Un paquet RPM source de expat est disponible sur <http://sourceforge.net/projects/expat/> .

8.141.3 Installation

Ces fonctions sont activées par défaut, et utilisent la bibliothèque expat fournie avec la distribution. Vous pouvez désactiver le support de XML en utilisant l'option de compilation --disable-xml . Si vous compilez PHP comme module pour Apache 1.3.9 ou supérieur, PHP va automatiquement utiliser la bibliothèque expat fournie par Apache. Si vous ne souhaitez pas utiliser la bibliothèque expat intégrée, il faut que vous compilez PHP avec l'option --with-expat-dir=DIR , où DIR est le dossier d'installation de votre bibliothèque expat.

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de bibliothèque supplémentaire pour disposer de ces fonctions.

8.141.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.141.5 Types de ressources

8.141.5.1 xml

La ressource `xml` est retournée par `xml_parser_create` et `xml_parser_create_ns`, et représente un analyseur XML à utiliser avec les autres fonctions de cette extension.

8.141.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

`XML_ERROR_NONE` (entier)
`XML_ERROR_NO_MEMORY` (entier)
`XML_ERROR_SYNTAX` (entier)
`XML_ERROR_NO_ELEMENTS` (entier)
`XML_ERROR_INVALID_TOKEN` (entier)
`XML_ERROR_UNCLOSED_TOKEN` (entier)
`XML_ERROR_PARTIAL_CHAR` (entier)
`XML_ERROR_TAG_MISMATCH` (entier)
`XML_ERROR_DUPLICATE_ATTRIBUTE` (entier)
`XML_ERROR_JUNK_AFTER_DOC_ELEMENT` (entier)
`XML_ERROR_PARAM_ENTITY_REF` (entier)
`XML_ERROR_UNDEFINED_ENTITY` (entier)
`XML_ERROR_RECURSIVE_ENTITY_REF` (entier)
`XML_ERROR_ASYNC_ENTITY` (entier)
`XML_ERROR_BAD_CHAR_REF` (entier)
`XML_ERROR_BINARY_ENTITY_REF` (entier)
`XML_ERROR_ATTRIBUTE_EXTERNAL_ENTITY_REF` (entier)
`XML_ERROR_MISPLACED_XML_PI` (entier)
`XML_ERROR_UNKNOWN_ENCODING` (entier)
`XML_ERROR_INCORRECT_ENCODING` (entier)
`XML_ERROR_UNCLOSED_CDATA_SECTION` (entier)
`XML_ERROR_EXTERNAL_ENTITY_HANDLING` (entier)
`XML_OPTION_CASE_FOLDING` (entier)
`XML_OPTION_TARGET_ENCODING` (entier)
`XML_OPTION_SKIP_TAGSTART` (entier)
`XML_OPTION_SKIP_WHITE` (entier)

8.141.7 gestionnaires d'événements

Les gestionnaires d'événements XML sont:

Fonction PHP de configuration du gestionnaire	Description de l'événement
<code>xml_set_element_handler</code>	Un événement est généré à chaque fois que l'analyseur XML rencontre une balise de début ou de fin. Deux gestionnaires sont disponibles : un pour le début, et un pour la fin.
<code>xml_set_character_data_handler</code>	

	"Character data" correspond grosso modo à tout ce qui n'est pas une balise XML, y compris les espaces entre les balises. Notez bien que l'analyseur XML n'ajoute ou n'efface aucun espace, et que c'est à l'application (c'est-à-dire vous) de décider de la signification de ces espaces.
<u>xml_set_processing_instruction_handler</u>	Les programmeurs PHP sont habitués aux instructions exécutables (processing instructions ou PIs). <code><?php ?></code> est une instruction exécutable où php est appelé programme cible. Ces instructions sont gérées de manière spécifique, (sauf le programme cible "XML" qui est réservé).
<u>xml_set_default_handler</u>	Tout ce qui n'a pas trouvé de gestionnaire est transmis au gestionnaire par défaut. Vous retrouverez par exemple, les déclarations de type de document dans ce gestionnaire.
<u>xml_set_unparsed_entity_decl_handler</u>	Ce gestionnaire est appelé pour gérer les déclarations des entités non analysées (NDATA).
<u>xml_set_notation_decl_handler</u>	Ce gestionnaire est appelé pour déclarer les notations.
<u>xml_set_external_entity_ref_handler</u>	Ce gestionnaire est appelé lorsque l'analyseur XML trouve une référence à une entité générale externe. Cela peut être une référence à un fichier ou à une URL. Reportez-vous à <u>entité externe</u> pour un exemple.

8.141.8 Problèmes de casse

Les fonctions de gestion des balises peuvent rencontrer des balises en minuscule, majuscule ou encore dans un mélange des deux. En XML, la procédure standard est d' "identifier les séquences de caractères qui ne sont pas reconnues comme majuscule, et de les remplacer par leur équivalent majuscule". En d'autres termes, XML met toutes lettres en majuscules.

Par défaut, tous les noms des éléments qui sont transmis aux fonctions de gestion sont mises en majuscule. Ce comportement est contrôlé par l'analyseur XML, et peut être lu et modifié avec les fonctions respectives xml_parser_get_option et xml_parser_set_option .

8.141.9 Codes d'erreurs

Les constantes suivantes sont définies comme des codes d'erreurs XML : (retournées par xml_parse)

- XML_ERROR_NONE
- XML_ERROR_NO_MEMORY
- XML_ERROR_SYNTAX
- XML_ERROR_NO_ELEMENTS
- XML_ERROR_INVALID_TOKEN
- XML_ERROR_UNCLOSED_TOKEN
- XML_ERROR_PARTIAL_CHAR
- XML_ERROR_TAG_MISMATCH
- XML_ERROR_DUPLICATE_ATTRIBUTE
- XML_ERROR_JUNK_AFTER_DOC_ELEMENT
- XML_ERROR_PARAM_ENTITY_REF
- XML_ERROR_UNDEFINED_ENTITY

- XML_ERROR_RECURSIVE_ENTITY_REF
- XML_ERROR_ASYNC_ENTITY
- XML_ERROR_BAD_CHAR_REF
- XML_ERROR_BINARY_ENTITY_REF
- XML_ERROR_ATTRIBUTE_EXTERNAL_ENTITY_REF
- XML_ERROR_MISPLACED_XML_PI
- XML_ERROR_UNKNOWN_ENCODING
- XML_ERROR_INCORRECT_ENCODING
- XML_ERROR_UNCLOSED_CDATA_SECTION
- XML_ERROR_EXTERNAL_ENTITY_HANDLING

8.141.10 Codage des caractères

L'extension XML de PHP supporte les caractères Unicode grâce à différents codages. Il y a deux types de codages de caractères : le codage à la source et le codage à la cible . PHP utilise le UTF-8 comme représentation interne.

L'encodage à la source est effectué lors de l'analyse du fichier par XML. Lors de la création d'un analyseur XML), un type de codage à la source peut être spécifié (et il ne pourra plus être modifié jusqu'à la destruction de l'analyseur). Les codages supportés sont : ISO-8859-1 , US-ASCII et UTF-8 . Les deux premiers sont des codages à un seul octet, c'est-à-dire que les caractères sont représentés sur un seul octet. UTF-8 peut représenter des caractères composés par un nombre variable de bits (jusqu'à 21), allant de 1 à quatre octets. Le codage par défaut utilisé par PHP ISO-8859-1 .

Le codage à la cible est effectué lorsque PHP transfère les données aux gestionnaires XML. Lorsqu'un analyseur est créé, le codage à la cible est spécifié de la même façon que le codage à la source, mais il peut être modifié à tout moment. Le codage à la cible affectera les balises, tout comme les données brutes, et les noms des instructions exécutables.

Si l'analyseur XML rencontre un caractère qu'il ne connaît pas (hors limite, par exemple), il retournera une erreur.

Si PHP rencontre un caractère dans le document XML analysé, qu'il ne peut pas représenter dans le codage à la cible choisi, le caractère sera remplacé par un point d'interrogation (cette attitude est susceptible de changer ultérieurement).

8.141.11 Exemples

Voici quelques exemples de code PHP analysant un document XML.

8.141.11.1 Exemple de structure XML

Ce premier exemple affiche la structure de l'élément de début dans un document avec indentation.

Afficher une structure XML
<pre><?php \$file = "donnees.xml"; \$depth = array(); function debutElement(\$parser, \$name, \$attrs) {</pre>

```

    global $depth;
    for ($i = 0; $i < $depth[$parser]; $i++) {
        echo " ";
    }
    echo "$name\n";
    $depth[$parser]++;
}

function finElement($parser, $name)
{
    global $depth;
    $depth[$parser]--;
}

$xml_parser = xml_parser_create();
xml_set_element_handler($xml_parser, "debutElement", "finElement");
if (!$fp = fopen($file, "r")) {
    die("Impossible d'ouvrir le fichier XML");
}

while ($data = fread($fp, 4096)) {
    if (!xml_parse($xml_parser, $data, feof($fp))) {
        die(sprintf("erreur XML : %s à la ligne %d",
            xml_error_string(xml_get_error_code($xml_parser)),
            xml_get_current_line_number($xml_parser)));
    }
}
xml_parser_free($xml_parser);
?>

```

8.141.11.2 Transtypage XML -> HTML

Transtypage XML -> HTML

Cet exemple remplace les balises XML d'un document par des balises HTML. Les éléments inconnus seront ignorés. Bien entendu, cet exemple sera appliqué à un type précis de fichiers XML.

```

<?php
$file = "data.xml";
$map_array = array(
    "BOLD" => "B",
    "EMPHASIS" => "I",
    "LITERAL" => "TT"
);

function startElement($parser, $name, $attrs)
{
    global $map_array;
    if (isset($map_array[$name])) {
        echo "<$map_array[$name]>";
    }
}

function endElement($parser, $name)
{
    global $map_array;
    if (isset($map_array[$name])) {
        echo "</$map_array[$name]>";
    }
}

function characterData($parser, $data)
{
    echo $data;
}

```

```

$xml_parser = xml_parser_create();
// Utilisons la gestion de casse, de manière à être sûrs de trouver la balise dans $map_array
xml_parser_set_option($xml_parser, XML_OPTION_CASE_FOLDING, true);
xml_set_element_handler($xml_parser, "startElement", "endElement");
xml_set_character_data_handler($xml_parser, "characterData");
if (!$fp = fopen($file, "r")) {
    die("Impossible de trouver le fichier XML");
}

while ($data = fread($fp, 4096)) {
    if (!xml_parse($xml_parser, $data, feof($fp))) {
        die(sprintf("erreur XML : %s à la ligne %d",
            xml_error_string(xml_get_error_code($xml_parser)),
            xml_get_current_line_number($xml_parser)));
    }
}
xml_parser_free($xml_parser);
?>

```

8.141.11.3 Entité externe

Cet exemple exploite les références externes de XML : il est possible d'utiliser un gestionnaire d'entité externe pour inclure et analyser les documents, tous comme les instructions exécutables peuvent servir à inclure et analyser d'autres documents, et aussi fournir une indication de confiance (voir plus bas).

Le document XML qui est utilisé dans cet exemple est fourni plus loin dans l'exemple (xmltest.xml et xmltest2.xml).

Entité externe

```

<?php
$file = "xmltest.xml";

function trustedFile($file)
{
    // only trust local files owned by ourselves
    if (!pregi("^([a-z]+)://", $file)
        && fileowner($file) == getmyuid()) {
        return true;
    }
    return false;
}

function startElement($parser, $name, $attribs)
{
    echo "<<font color=\">#0000cc\">$name</font>";
    if (count($attribs)) {
        foreach ($attribs as $k => $v) {
            echo " <font color=\">#009900\ ">$k</font>=<font
                color=\">#990000\ ">$v</font><";
        }
    }
    echo ">";
}

function endElement($parser, $name)
{
    echo "<</font color=\">#0000cc\ ">$name</font>>";
}

function characterData($parser, $data)

```

```

{
    echo "<strong>$data</strong>";
}

function PIHandler($parser, $target, $data)
{
    switch (strtolower($target)) {
        case "php":
            global $parser_file;
            // If the parsed document is "trusted", we say it is safe
            // to execute PHP code inside it. If not, display the code
            // instead.
            if (trustedFile($parser_file[$parser])) {
                eval($data);
            } else {
                printf("Untrusted PHP code: <em>%s</em>",
                    htmlspecialchars($data));
            }
            break;
    }
}

function defaultHandler($parser, $data)
{
    if (substr($data, 0, 1) == "&" && substr($data, -1, 1) == ";") {
        printf('<font color="#aa00aa">%s</font>',
            htmlspecialchars($data));
    } else {
        printf('<font size="-1">%s</font>',
            htmlspecialchars($data));
    }
}

function externalEntityRefHandler($parser, $openEntityNames, $base, $systemId,
    $publicId) {
    if ($systemId) {
        if (!list($parser, $fp) = new_xml_parser($systemId)) {
            printf("Could not open entity %s at %s\n", $openEntityNames,
                $systemId);
            return false;
        }
        while ($data = fread($fp, 4096)) {
            if (!xml_parse($parser, $data, feof($fp))) {
                printf("erreur XML : %s à la ligne %d lors de l'analyse de l'entité %s\n",
                    xml_error_string(xml_get_error_code($parser)),
                    xml_get_current_line_number($parser), $openEntityNames);
                xml_parser_free($parser);
                return false;
            }
        }
        xml_parser_free($parser);
        return true;
    }
    return false;
}

function new_xml_parser($file)
{
    global $parser_file;

    $xml_parser = xml_parser_create();
    xml_parser_set_option($xml_parser, XML_OPTION_CASE_FOLDING, 1);
    xml_set_element_handler($xml_parser, "startElement", "endElement");
    xml_set_character_data_handler($xml_parser, "characterData");
    xml_set_processing_instruction_handler($xml_parser, "PIHandler");
    xml_set_default_handler($xml_parser, "defaultHandler");
    xml_set_external_entity_ref_handler($xml_parser, "externalEntityRefHandler");
}

```

```

    if (!($fp = @fopen($file, "r"))) {
        return false;
    }
    if (!is_array($parser_file)) {
        settype($parser_file, "array");
    }
    $parser_file[$xml_parser] = $file;
    return array($xml_parser, $fp);
}

if (!(list($xml_parser, $fp) = new_xml_parser($file))) {
    die("Impossible d'ouvrir le fichier XML");
}

echo "<pre>";
while ($data = fread($fp, 4096)) {
    if (!xml_parse($xml_parser, $data, feof($fp))) {
        die(sprintf("Erreur XML : %s à la ligne %d\n",
            xml_error_string(xml_get_error_code($xml_parser)),
            xml_get_current_line_number($xml_parser)));
    }
}
echo "</pre>";
echo "parse complete\n";
xml_parser_free($xml_parser);
?>

```

xmltest.xml

```

<!DOCTYPE chapter SYSTEM "/just/a/test.dtd" [
<!ENTITY plainEntity "FOO entity">
<!ENTITY systemEntity SYSTEM "xmltest2.xml">
]>
<chapter>
<TITLE>Title </TITLE>
<para>
<informaltable>
<tgroup cols="3">
<tbody>
<row><entry>a1</entry><entry morerows="1">b1</entry><entry>c1</entry></row>
<row><entry>a2</entry><entry>c2</entry></row>
<row><entry>a3</entry><entry>b3</entry><entry>c3</entry></row>
</tbody>
</tgroup>
</informaltable>
</para>

<section id="about">
<title>About this Document</title>
<para>

<?php print 'Salut! Ceci est PHP version '.phpversion(); ?>
</para>
</section>
</chapter>

```

Ce fichier est inclus depuis xmltest.xml :

xmltest2.xml

```

<!DOCTYPE foo [
<!ENTITY testEnt "test entity">
]>
<foo>

```

```

<element attrib="value"/>

<?php print "Ceci est du code PHP qui est exécuté."; ?>
</foo>

```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [gestionnaires d'événements](#)
- [Problèmes de casse](#)
- [Codes d'erreurs](#)
- [Codage des caractères](#)
- [Exemples](#)
- [utf8_decode](#)
- [utf8_encode](#)
- [xml_error_string](#)
- [xml_get_current_byte_index](#)
- [xml_get_current_column_number](#)
- [xml_get_current_line_number](#)
- [xml_get_error_code](#)
- [xml_parse_into_struct](#)
- [xml_parse](#)
- [xml_parser_create_ns](#)
- [xml_parser_create](#)
- [xml_parser_free](#)
- [xml_parser_get_option](#)
- [xml_parser_set_option](#)
- [xml_set_character_data_handler](#)
- [xml_set_default_handler](#)
- [xml_set_element_handler](#)
- [xml_set_end_namespace_decl_handler](#)
- [xml_set_external_entity_ref_handler](#)
- [xml_set_notation_decl_handler](#)
- [xml_set_object](#)
- [xml_set_processing_instruction_handler](#)
- [xml_set_start_namespace_decl_handler](#)
- [xml_set_unparsed_entity_decl_handler](#)

8.141.13 utf8_encode() : Convertit une chaîne ISO-8859-1 en UTF-8

string **utf8_encode** (string data)

`utf8_encode` code la chaîne data au format UTF-8 , et retourne la version codée. UTF-8 est un mécanisme standardisé utilisé par Unicode pour coder les caractère de grande taille dans des flux d'octets. UTF-8 est transparent pour les caractères *ASCII* , il est auto-synchronisé (c'est à dire qu'un programme peut toujours savoir dans un flux d'octet où un caractère commence), et peut être utilisé pour faire des comparaisons de chaînes standard, comme pour le tri. PHP utilise l' UTF-8 pour coder les caractères jusqu'à 4 octets comme ceci :

octets	bits	représentation
--------	------	----------------

1	7	0bbbbbbb
2	11	110bbbb 10bbbbbb
3	16	1110bbbb 10bbbbbb 10bbbbbb
4	21	11110bbb 10bbbbbb 10bbbbbb 10bbbbbb

Chaque b représente un bit qui peut être utilisé pour enregistrer un caractère.

8.141.14 `xml_error_string()` : Lit le message d'erreur de l'analyseur XML

string `xml_error_string` (int `code`)

Lit le message d'erreur de l'analyseur XML associé avec la code `code` donné.

8.141.15 `xml_get_current_byte_index()` : Retourne l'index de l'octet courant d'un analyseur XML

int `xml_get_current_byte_index` (resource `parser`)

Retourne l'index de l'octet courant de l'analyseur XML donné.

8.141.16 `xml_get_current_column_number()` : Retourne le nombre courant de colonne d'un analyseur XML

int `xml_get_current_column_number` (resource `parser`)

Retourne le nombre courant de colonne de l'analyseur XML donné.

8.141.17 `xml_get_current_line_number()` : Retourne le numéro de ligne courant d'un analyseur XML

int `xml_get_current_line_number` (resource `parser`)

Retourne le numéro de ligne courant de l'analyseur XML donné.

8.141.18 `xml_get_error_code()` : Récupère le code erreur de l'analyseur XML

int `xml_get_error_code` (resource `parser`)

Récupère le code erreur de l'analyseur XML.

8.141.19 `xml_parse_into_struct()` : Analyse une structure XML

int `xml_parse_into_struct` (resource `parser` , string `data` , array `values` , array `index`)

`xml_parse_into_struct` analyse le fichier XML `data`, et le place dans deux tableaux : le premier index contient des pointeurs sur la position des valeurs correspondantes dans le tableau `values`. Ces deux paramètres sont passés par références.

Note

`xml_parse_into_struct` retourne 0 si une erreur survient et 1 en cas de succès. Ce n'est pas la même chose que FALSE et TRUE, soyez prudent avec les opérateurs comme `===`.

Ci-dessous, vous trouverez un exemple qui illustre la structure des deux tableaux générés par la fonction. On utilise une balise simple `note`, placée dans une autre balise `para`. On analyse le tout, et on affiche la structure générée :

Exemple avec `xml_parse_into_struct`

```
<?php
$simple = "<para><note>Simple Note</note></para>";
$p = xml_parser_create();
xml_parse_into_struct($p, $simple, $vals, $index);
xml_parser_free($p);
echo "Tableau d'index \n";
print_r($index);
echo "\nTableau de valeurs \n";
print_r($vals);
?>
```

L'exemple ci-dessus va afficher :

Tableau d'index

```
Array
(
  [PARA] => Array
  (
    [0] => 0
    [1] => 2
  )
  [NOTE] => Array
  (
    [0] => 1
  )
)
```

Tableau de valeurs

```
Array
(
  [0] => Array
  (
    [tag] => PARA
    [type] => open
    [level] => 1
  )
  [1] => Array
  (
    [tag] => NOTE
```

```

[type] => complete
[level] => 2
[value] => Simple Note
)

[2] => Array
(
[tag] => PARA
[type] => close
[level] => 1
)
)

```

L'analyse événementielle (comme celle de expat), peut se révéler complexe lorsque le document XML est complexe. `xml_parse_into_struct` ne génère pas d'objet de type DOM, mais il génère plutôt des structures qui peuvent être parcourues à la façon d'un arbre. Considérons le fichier suivant, qui représente une petite base de données XML :

molddb.xml - Petite base de données moléculaires

```

<molddb>

  <molecule>
    <name>Alanine</name>
    <symbol>ala</symbol>
    <code>A</code>
    <type>hydrophobic</type>
  </molecule>

  <molecule>
    <name>Lysine</name>
    <symbol>lys</symbol>
    <code>K</code>
    <type>charged</type>
  </molecule>

</molddb>

```

Et maintenant, un code qui analyse le document, et génère les objets ad hoc :

parsemolddb.php : Analyse molddb.xml et crée un tableau d'objets moléculaires

```

<?php

class AminoAcid {
    var $name; // nom aa
    var $symbol; // symbole à trois lettres
    var $code; // code à une lettre
    var $type; // hydrophobique, chargé ou neutre

    function AminoAcid ($aa) {
        foreach ($aa as $k=>$v)
            $this->$k = $aa[$k];
    }
}

function readDatabase($filename) {
    // read the xml database of aminoacids
    $data = implode("", file($filename));
    $parser = xml_parser_create();
    xml_parser_set_option($parser, XML_OPTION_CASE_FOLDING, 0);
    xml_parser_set_option($parser, XML_OPTION_SKIP_WHITE, 1);
    xml_parse_into_struct($parser, $data, $values, $tags);
    xml_parser_free($parser);
}

```

```

// loop through the structures
foreach ($tags as $key=>$val) {
    if ($key == "molecule") {
        $molranges = $val;
        // each contiguous pair of array entries are the
        // lower and upper range for each molecule definition
        for ($i=0; $i < count($molranges); $i+=2) {
            $offset = $molranges[$i] + 1;
            $len = $molranges[$i + 1] - $offset;
            $tdb[] = parseMol(array_slice($values, $offset, $len));
        }
    } else {
        continue;
    }
}
return $tdb;
}

function parseMol($mvalues) {
    for ($i=0; $i < count($mvalues); $i++)
        $mol[$mvalues[$i]["tag"]] = $mvalues[$i]["value"];
    return new AminoAcid($mol);
}

$db = readDatabase("molddb.xml");
echo "*** Base d'objets AminoAcid :\n";
print_r($db);

?>

```

Après exécution de parsemolddb.php , la variable \$db contient un tableau d'objets AminoAcid , et l'affichage le confirme :

```

** Base d'objets AminoAcid :
Array
(
    [0] => aminoacid Object
    (
        [name] => Alanine
        [symbol] => ala
        [code] => A
        [type] => hydrophobic
    )

    [1] => aminoacid Object
    (
        [name] => Lysine
        [symbol] => lys
        [code] => K
        [type] => charged
    )
)

```

8.141.20 xml_parse() : Commence l'analyse d'un document XML

int **xml_parse** (resource parser , string data , bool is_final)

xml_parse analyse un document XML. Les gestionnaires pour les événements configurés sont appelés autant de fois que nécessaire.

8.141.21 xml_parser_create_ns() : Crée un analyseur XML

resource xml_parser_create_ns (*string encoding* , *string separator*)

xml_parser_create_ns crée un nouvel analyseur XML avec le support des espaces de noms, et retourne une ressource, qui pourra être utilisée avec les autres fonctions XML.

Avec un analyseur qui supporte les espaces de noms, les balises qui sont passées aux différentes fonctions de gestion seront constituées du nom de l'espace et du nom de balise, séparé par la chaîne separator ou par ':' , par défaut.

Le paramètre optionnel encoding spécifie le jeu de caractères d'encodage pour l'entrée/sortie dans PHP 4. Depuis PHP 5, ce jeu de caractères est automatiquement détecté, et donc, le paramètre encoding ne spécifie plus que la sortie. En PHP 4, le jeu de caractères de sortie par défaut est le même que celui d'entrée. En PHP 5.0.0 et PHP 5.0.1, le jeu de caractères d'entrée par défaut est ISO-8859-1, tandis qu'en PHP 5.0.2 et suivant, il vaut UTF-8. Les jeux de caractères supportés sont ISO-8859-1 , UTF-8 et US-ASCII .

Voir aussi xml_parser_create et xml_parser_free .

8.141.22 xml_parser_create() : Création d'un analyseur XML

resource xml_parser_create (*string encoding*)

xml_parser_create crée un analyseur XML et retourne une référence sur cet analyseur pour qu'il puisse être utilisé ultérieurement par d'autres fonctions XML.

Le paramètre optionnel encoding spécifie le jeu de caractères d'encodage pour l'entrée/sortie dans PHP 4. Depuis PHP 5, ce jeu de caractères est automatiquement détecté, et donc, le paramètre encoding ne spécifie plus que la sortie. En PHP 4, le jeu de caractères de sortie par défaut est le même que celui d'entrée. Si une chaîne vide est passée, le parseur tente d'identifier quel jeu de caractère a été utilisé pour encoder le document en regardant les 3 ou 4 octets du haut. En PHP 5.0.0 et PHP 5.0.1, le jeu de caractères d'entrée par défaut est ISO-8859-1, tandis qu'en PHP 5.0.2 et suivant, il vaut UTF-8. Les jeux de caractères supportés sont ISO-8859-1 , UTF-8 et US-ASCII .

Voir aussi xml_parser_create_ns et xml_parser_free .

8.141.23 xml_parser_free() : Détruit un analyseur XML

bool xml_parser_free (resource parser)

parser

Une référence sur un analyseur XML.

xml_parser_free retourne FALSE si parser n'est pas une référence valide, ou sinon, détruit l'analyseur et retourne TRUE .

8.141.24 `xml_parser_get_option()` : Lit les options d'un analyseur XML

mixed `xml_parser_get_option` (resource parser , int option)

parser

Une référence sur un analyseur XML valide.

option

L'option demandée. `XML_OPTION_CASE_FOLDING` et `XML_OPTION_TARGET_ENCODING` sont disponibles. Reportez-vous à [xml_parser_set_option](#) pour leurs descriptions.

`xml_parser_get_option` retourne `FALSE` si `parser` n'est pas un parseur valide ou si `option` n'est pas valide (genère aussi un `E_WARNING`). Sinon, elle retourne la valeur de l'option demandée.

8.141.25 `xml_parser_set_option()` : Affecte les options d'un analyseur XML

bool `xml_parser_set_option` (resource parser , int option , mixed value)

parser

Une référence vers un analyseur XML.

option

L'option à modifier. Voir ci-dessous :

value

La nouvelle valeur de l'option.

`xml_parser_set_option` retourne `FALSE` si `parser` n'est pas une référence valide sur un analyseur XML, ou si l'option n'a pas pu être modifiée. Sinon, l'option est effectivement modifiée, et la fonction retourne `TRUE` .

Les options suivantes sont disponibles :

Option	Type de données	Description
<code>XML_OPTION_CASE_FOLDING</code>	entier	Contrôle la gestion de la <u>cas</u> se des balises de cet analyseur XML. Par défaut, activé.
<code>XML_OPTION_SKIP_TAGSTART</code>	integer	Spécifie combien de caractères doivent être éludés du début du nom de la balise.
<code>XML_OPTION_SKIP_WHITE</code>	integer	Élude ou non les valeurs contenant des caractères blancs.
<code>XML_OPTION_TARGET_ENCODING</code>	chaîne de caractères	Modifie le <u>codage à la cible</u> utilisé par cet analyseur XML. Par défaut, c'est celui qui a été spécifié lors de l'appel de <code>xml_parser_create</code> . Les codages supportés sont ISO-8859-1 , US-ASCII et UTF-8 .

8.141.26 `xml_set_character_data_handler()` : Affecte les gestionnaires de texte littéral

bool `xml_set_character_data_handler` (resource parser , callback handler)

xml_set_character_data_handler affecte les gestionnaires de début et de fin de l'analyseur XML parser . handler est une chaîne qui contient le nom d'une fonction qui existe lorsque xml_parse est appelé pour créer parser .

La fonction handler doit accepter deux paramètres : **handler** resource parser (string data)

parser

Le premier paramètre, parser , est une référence sur l'analyseur XML qui appelle cette fonction.

data

Le second paramètre, data , contient les caractères sous la forme d'une chaîne.

Si un gestionnaire reçoit une chaîne vide ou FALSE , c'est qu'il est désactivé.

xml_set_character_data_handler retourne TRUE si le gestionnaire est actif, et FALSE sinon, ou si parser n'est pas un analyseur.

Note

A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé.

8.141.27 xml_set_default_handler() : Affecte le gestionnaire XML par défaut

bool xml_set_default_handler (resource parser , callback handler)

xml_set_default_handler affecte le gestionnaire par défaut de l'analyseur XML parser . handler est une chaîne qui contient le nom d'une fonction qui existe lorsque xml_parse est appelé pour créer parser .

La fonction handler doit accepter deux paramètres: **handler** resource parser (string data)

parser

Le premier paramètre, parser , est une référence sur l'analyseur XML qui appelle cette fonction.

data

Le second paramètre, data , contient les caractères sous la forme d'une chaîne. Cela peut être une déclaration XML, un type de document, une entité ou d'autres données pour qui aucun gestionnaire n'est prévu.

Si un gestionnaire reçoit une chaîne vide ou FALSE , c'est qu'il est désactivé.

xml_set_default_handler retourne TRUE si le gestionnaire est actif, et FALSE sinon, ou si parser n'est pas un analyseur.

Note

A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé.

8.141.28 xml_set_element_handler() : Affecte les gestionnaires de début et de fin de balise XML

bool **xml_set_element_handler** (resource parser , callback start_element_handler , callback end_element_handler)

xml_set_element_handler affecte les gestionnaires de début et de fin de l'analyseur XML parser . start_element_handler et end_element_handler sont des chaînes qui contiennent les noms de fonctions qui existent lorsque xml_parse est appelé pour créer parser .

La fonction start_element_handler doit accepter trois paramètres: **start_element_handler** resource parser (string name , array attrs)

parser

Le premier paramètre, parser , est une référence sur l'analyseur XML qui appelle cette fonction.

name

Le deuxième paramètre, name , contient le nom de l'élément qui a provoqué l'appel du gestionnaire. Si l'analyseur gère la casse , cet élément sera en majuscule.

attrs

Le troisième paramètre, attrs , contient un tableau associatif avec les attributs de l'éléments (s'il en existe). Les clés de ce tableau seront les noms des attributs, et les valeurs seront les valeurs correspondantes des attributs. Les noms des attributs seront mis en majuscule si l'analyseur gère la casse . Les valeurs des attributs seront intouchées.

L'ordre original des attributs peut être retrouvé en passant en revue le tableau attrs , avec la fonction each . La première clé sera la première clé du tableau.

La fonction endelementhandler doit accepter deux paramètres: **end_element_handler** resource parser (string name)

parser

Le premier paramètre, parser , est une référence sur l'analyseur XML qui appelle cette fonction.

name

Le second paramètre, name , contient le nom de l'élément qui a provoqué l'appel du gestionnaire. Si l'analyseur gère la casse , cet élément sera en majuscule.

Si un gestionnaire reçoit une chaîne vide, ou FALSE , c'est qu'il est en train d'être désactivé.

xml_set_element_handler retourne TRUE si le gestionnaire est actif, et FALSE sinon, ou si parser n'est pas un analyseur.

Note

A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé.

8.141.29 **xml_set_end_namespace_decl_handler()** : Configure le gestionnaire XML de données

bool **xml_set_end_namespace_decl_handler** (resource parser , callback handler)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Note

8.141.28 **xml_set_element_handler()** : Affecte les gestionnaires de début et de fin de balise XML2083

A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé.

8.141.30 `xml_set_external_entity_ref_handler()` : Configure le gestionnaire XML de références externes

`bool xml_set_external_entity_ref_handler (resource parser , callback handler)`

`xml_set_external_entity_ref_handler` affecte le gestionnaire d'entité externe de l'analyseur XML `parser` . `handler` et `endelementhandler` sont des chaînes qui contiennent les noms de fonction qui existent lorsque `xml_parse` est appelé pour créer le `parser` .

La fonction `handler` doit accepter 5 paramètres, et retourner un entier. Si la valeur retournée par le gestionnaire est `FALSE` (par exemple si aucune valeur n'est retournée), l'analyseur XML s'arrêtera, et la fonction `xml_get_error_code` retournera `XML_ERROR_EXTERNAL_ENTITY_HANDLING` .
handler `resource parser (string open_entity_names , string base , string system_id , string public_id)`

`parser`

Le premier paramètre, `parser` , est une référence sur l'analyseur XML qui appelle cette fonction.

`open_entity_names`

Le deuxième paramètre, `open_entity_names` , est la liste de noms d'entité, séparés par des espaces. Ces entités sont accessibles à l'analyse par cet entité (y compris le nom de l'entité référencé).

`base`

La meilleure base de résolution de l'identifiant système (`system_id`) de cet entité externe. Actuellement, ce paramètre est toujours une chaîne vide.

`system_id`

Identifiant système pour cet entité externe.

`public_id`

Le cinquième paramètre, `public_id` , est l'identifiant public, comme spécifié dans la déclaration d'entité, ou un chaîne vide, si aucune déclaration n'a été spécifiée. L'espace dans l'identifiant public sera normalisé comme spécifié dans les spécifications XML.

Si un gestionnaire reçoit une chaîne vide, ou `FALSE` , c'est qu'il est en train d'être désactivé.

`xml_set_external_entity_ref_handler` retourne `TRUE` si le gestionnaire est actif, et `FALSE` sinon ou si `parser` n'est pas un analyseur.

Note

A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé.

8.141.31 `xml_set_notation_decl_handler()` : Configure le gestionnaire XML de notations

`bool xml_set_notation_decl_handler (resource parser , callback handler)`

`xml_set_notation_decl_handler` affecte les gestionnaires de début et de fin de l'analyseur XML `parser` . `handler` est une chaîne qui contient le nom d'une fonction qui existe lorsque `xml_parse` est appelé pour créer `parser` .

8.141.29 `xml_set_end_namespace_decl_handler()` : Configure le gestionnaire XML de données 2084

Une notation est une partie du DTD du document, qui a le format suivant :

```
<!NOTATION <parameter>name</parameter>
{ <parameter>systemid</parameter> | <parameter>publicid</parameter>?}>
```

Reportez-vous à la section [des spécifications XML 1.0](#) pour connaître les notations des entités externes.

La fonction handler doit accepter cinq paramètres : **handler** resource parser (string notation_name , string base , string system_id , string public_id)

parser

Le premier paramètre, parser , est une référence sur l'analyseur XML qui appelle cette fonction.

notation_name

Le nom de la notation, name , comme précisé dans le format de notation ci-dessus.

base

La meilleure base de résolution de l'identifiant système (system_id) de cette entité externe. Actuellement, ce paramètre est toujours une chaîne vide.

system_id

Identifiant système pour cet entité externe.

public_id

Identifiant public pour cet entité externe.

Si un gestionnaire reçoit une chaîne vide ou FALSE , c'est qu'il est en train d'être désactivé.

[xml_set_notation_decl_handler](#) retourne TRUE si le gestionnaire est actif, et FALSE sinon ou si parser n'est pas un analyseur.

Note

A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé.

8.141.32 [xml_set_object\(\)](#) : Utilise un analyseur XML à l'intérieur d'un objet

bool [xml_set_object](#) (resource parser , object object)

[xml_set_object](#) rend l'analyseur parser utilisable depuis un objet. Toutes les méthodes de callback, affectées par [xml_set_element_handler](#) , seront les méthodes de cet objet.

Exemple avec [xml_set_object](#)

```
<?php
class xml {
    var $parser;

    function xml()
    {
        $this->parser = xml_parser_create();

        xml_set_object($this->parser, $this);
        xml_set_element_handler($this->parser, "tag_open", "tag_close");
        xml_set_character_data_handler($this->parser, "cdata");
    }

    function parse($data)
```

```

    {
        xml_parse($this->parser, $data);
    }

    function tag_open($parser, $tag, $attributes)
    {
        var_dump($parser, $tag, $attributes);
    }

    function cdata($parser, $cdata)
    {
        var_dump($parser, $cdata);
    }

    function tag_close($parser, $tag)
    {
        var_dump($parser, $tag);
    }
} // end of class xml

$xml_parser = new xml();
$xml_parser->parse("<a id='hallo'>PHP</a>");
?>

```

8.141.33 `xml_set_processing_instruction_handler()` : Affecte les gestionnaires d'instructions exécutables

`bool xml_set_processing_instruction_handler (resource parser , callback handler)`

`xml_set_processing_instruction_handler` affecte le gestionnaire d'instructions exécutables de l'analyseur XML `parser`. `handler` est une chaîne qui contient le nom d'une fonction qui existe lorsque `xml_parse` est appelé pour créer `parser`.

Une instruction exécutable a la forme suivante :

Structure d'instruction exécutable
<pre> <? target data ?> </pre>

Vous pouvez mettre du code PHP entre ces balises, mais soyez conscient d'une des limitations des instructions exécutables de XML : la balise de fin d'instruction exécutable (`?>`) ne peut être échappée, ce qui fait que cette séquence NE DOIT JAMAIS apparaître dans le code PHP placé dans le document PHP. Si un tel texte apparaît, la balise de fin d'instruction exécutable sera reconnue, et le reste du code sera considéré comme des données brutes (et donc, pas exécutées).

La fonction handler doit accepter trois paramètres: **handler** `resource parser (string target , string data)`

`parser`

Le premier paramètre, `parser`, est une référence sur l'analyseur XML qui appelle cette fonction.

`target`

Le second paramètre, `target`, contient l'application cible.

`data`

Le troisième paramètre, data , contient le code sous la forme d'une chaîne.

Si un gestionnaire reçoit une chaîne vide, ou FALSE , c'est qu'il est désactivé.

`xml_set_processing_instruction_handler` retourne TRUE si le gestionnaire est actif, et FALSE sinon, ou si parser n'est pas un analyseur.

Note
A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé.

8.141.34 `xml_set_start_namespace_decl_handler()` : Configure le gestionnaire de caractères

`bool xml_set_start_namespace_decl_handler (resource parser , callback handler)`

Attention
Cette fonction n'est pas documentée et seule la liste des arguments est disponible.
Note
A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé.

8.141.35 `xml_set_unparsed_entity_decl_handler()` : Affecte les gestionnaires d'entités non déclarées

`bool xml_set_unparsed_entity_decl_handler (resource parser , callback handler)`

`xml_set_unparsed_entity_decl_handler` affecte les gestionnaires d'entités non déclarées de l'analyseur XML parser . handler est une chaîne qui contient le nom d'une fonction qui existe lorsque `xml_parse` est appelé pour créer parser .

Ce gestionnaire sera appelé si l'analyseur XML rencontre une déclaration d'entité externe avec une déclaration de NDATA, comme suit :

```
<!ENTITY <parameter>name</parameter> {<parameter>publicid</parameter> | <parameter>systemid</parameter>
      NDATA <parameter>notationname</parameter>
```

Reportez-vous à la section [des spécifications XML 1.0](#) pour connaître les notations des entités externes.

La fonction handler doit accepter six paramètres: **handler** resource parser (string entity_name , string base , string system_id , string public_id , string notation_name)

- parser
Le premier paramètre, parser , est une référence sur l'analyseur XML qui appelle cette fonction.
- entity_name
Le nom de l'entité qui va être définie
- base
La meilleure base de résolution de l'identifiant système de cette entité externe. Actuellement, ce paramètre est toujours une chaîne vide.
- system_id

Identifiant système pour cet entité externe.

`public_id`

Identifiant public pour cet entité externe.

`notationname`

Nom de la notation de cette entité. (Voir [xml_set_notation_decl_handler](#)).

Si un gestionnaire reçoit une chaîne vide ou FALSE , c'est qu'il est en train d'être désactivé.

[xml_set_unparsed_entity_decl_handler](#) retourne TRUE si le gestionnaire est actif, et FALSE sinon, ou si parser n'est pas un analyseur.

Note

A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé.

8.142 XML-RPC

8.142.1 Introduction

Ces fonctions servent à écrire des serveurs et clients XML-RPC. Vous trouverez plus d'informations sur XML-RPC sur le site <http://www.xmlrpc.com/>, et plus de documentation sur cette extension et ses fonctions à <http://xmlrpc-epi.sourceforge.net/>.

Attention
Ce module est EXPERIMENTAL . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez ce module à vos risques et périls.

8.142.2 Pré-requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

8.142.3 Installation

Le support de XML-RPC en PHP n'est pas activé par défaut. Vous avez besoin de l'activer grâce à l'option de compilation `--with-xmlrpc[=DIR]`. Cette extension est fournie avec PHP depuis la version 4.1.0.

8.142.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier `php.ini`.

Nom	Par défaut	Modifiable	Historique
<code>xmlrpc_errors</code>	"0"	PHP_INI_SYSTEM	Disponible depuis PHP 4.1.0.
<code>xmlrpc_error_number</code>	"0"	PHP_INI_ALL	Disponible depuis PHP 4.1.0.

Pour plus de détails sur les constantes `PHP_INI_*`, reportez-vous à [Directives du php.ini](#).

8.142.5 Types de ressources

Cette extension ne définit aucune ressource.

8.142.6 Constantes pré-définies

Cette extension ne définit aucune constante.

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)

- [xmlrpc_decode_request](#)
- [xmlrpc_decode](#)
- [xmlrpc_encode_request](#)
- [xmlrpc_encode](#)
- [xmlrpc_get_type](#)
- [xmlrpc_is_fault](#)
- [xmlrpc_parse_method_descriptions](#)
- [xmlrpc_server_add_introspection_data](#)
- [xmlrpc_server_call_method](#)
- [xmlrpc_server_create](#)
- [xmlrpc_server_destroy](#)
- [xmlrpc_server_register_introspection_callback](#)
- [xmlrpc_server_register_method](#)
- [xmlrpc_set_type](#)

8.142.8 xmlrpc_decode() : Décode le code XML en types PHP natifs

array **xmlrpc_decode** (string *xml* , string *encoding*)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.142.9 xmlrpc_encode_request() : Génère le code XML d'une requête de méthode

string **xmlrpc_encode_request** (string *method* , mixed *params* , array *output_options*)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.142.10 xmlrpc_encode() : Génère le code XML pour une valeur PHP

string **xmlrpc_encode** (mixed *value*)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.142.11 `xmlrpc_get_type()` : Retourne le type XMLRPC d'une valeur PHP

string `xmlrpc_get_type` (mixed value)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Particulièrement pratique pour les types chaînes de type base64 et datetime.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.142.12 `xmlrpc_is_fault()` : Détermine si une valeur de tableau représente un XMLRPC

bool `xmlrpc_is_fault` (array arg)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.142.13 `xmlrpc_parse_method_descriptions()` : Décode le code XML en une liste de descriptions de méthodes

array `xmlrpc_parse_method_descriptions` (string xml)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.142.14 `xmlrpc_server_add_introspection_data()` : Ajoute des données d'introspection

int `xmlrpc_server_add_introspection_data` (resource server , array desc)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS

PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.142.15 `xmlrpc_server_call_method()` : Analyse une requête XML est appelle les méthodes associées

string `xmlrpc_server_call_method` (resource `server` , string `xml` , mixed `user_data` , array `output_options`)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.142.16 `xmlrpc_server_create()` : Crée un serveur XMLRPC

resource `xmlrpc_server_create` (void)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.142.17 `xmlrpc_server_destroy()` : Détruit un serveur XMLRPC

int `xmlrpc_server_destroy` (resource `server`)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.142.18 `xmlrpc_server_register_introspection_callback()` : Enregistre une fonction PHP pour générer la documentation

bool `xmlrpc_server_register_introspection_callback` (resource `server` , string `fonction`)

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.142.19 `xmlrpc_server_register_method()` : Enregistre une fonction PHP avec une méthode

`bool xmlrpc_server_register_method (resource server , string method_name , string fonction)`

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.142.20 `xmlrpc_set_type()` : Modifie le type XMLRPC, base64 ou datetime, pour une chaîne de caractères PHP

`bool xmlrpc_set_type (string value , string type)`

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.143 XSL

8.143.1 Introduction

L'extension XSL implémente le standard XSL, et fait des transformations XSLT à l'aide de la bibliothèque libxslt .

8.143.2 Pré-requis

Cette extension utilise la bibliothèque libxslt qui peut être trouvée sur <http://xmlsoft.org/XSLT/> . La bibliothèque libxslt version 1.0.18 ou supérieure est requise.

8.143.3 Installation

PHP 5 inclut l'extension XSL par défaut et peut être activée en ajoutant l'argument `--with-xsl[=DIR]` à votre ligne de configuration. DIR est le dossier d'installation de la bibliothèque libxslt .

8.143.4 Classes pré-définies

8.143.4.1 XSLTProcessor

8.143.4.1.1 Constructeur

- XSLTProcessor->__construct() - construit un nouvel objet XSLTProcessor

8.143.4.1.2 Méthodes

- XSLTProcessor->getParameter() - Récupère la valeur d'un paramètre
- XSLTProcessor->hasExsltSupport() - Détermine si PHP a le support EXSLT
- XSLTProcessor->importStylesheet() - Importe une feuille de style
- XSLTProcessor->registerPHPFunctions() - Active la possibilité d'utiliser les fonctions PHP comme fonctions XSLT
- XSLTProcessor->removeParameter() - Efface un paramètre
- XSLTProcessor->setParameter() - Définit la valeur d'un paramètre
- XSLTProcessor->transformToDoc() - Transforme en un DOMDocument
- XSLTProcessor->transformToURI() - Transforme en une URI
- XSLTProcessor->transformToXML() - Transforme en du XML

8.143.5 Exemples

Plusieurs exemples dans ce manuel requiert un fichier XML ainsi qu'un fichier XSL. Nous utiliserons `collection.xml` et `collection.xsl` qui contiennent ceci :

```
collection.xml
<collection>
```

```

<cd>
  <title>Fight for your mind</title>
  <artist>Ben Harper</artist>
  <year>1995</year>
</cd>
<cd>
  <title>Electric Ladyland</title>
  <artist>Jimi Hendrix</artist>
  <year>1997</year>
</cd>
</collection>

```

collection.xsl

```

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:param name="owner" select="'Nicolas Eliazewicz'"/>
  <xsl:output method="html" encoding="iso-8859-1" indent="no"/>
  <xsl:template match="collection">
    Hey! Welcome to <xsl:value-of select="$owner"/>'s sweet CD collection!
  <xsl:apply-templates/>
</xsl:template>
  <xsl:template match="cd">
    <h1><xsl:value-of select="title"/></h1>
    <h2>by <xsl:value-of select="artist"/> - <xsl:value-of select="year"/></h2>
    <hr />
  </xsl:template>
</xsl:stylesheet>

```

8.143.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

XSL_CLONE_AUTO ([entier](#))
 XSL_CLONE_NEVER ([entier](#))
 XSL_CLONE_ALWAYS ([entier](#))

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Classes pré-définies](#)
- [Exemples](#)
- [Constantes pré-définies](#)
- [XSLTProcessor->__construct\(\)](#)
- [XSLTProcessor->getParameter\(\)](#)
- [XSLTProcessor->hasExsltSupport\(\)](#)
- [XSLTProcessor->importStylesheet\(\)](#)
- [XSLTProcessor->registerPHPFunctions\(\)](#)
- [XSLTProcessor->removeParameter\(\)](#)
- [XSLTProcessor->setParameter\(\)](#)
- [XSLTProcessor->transformToDoc\(\)](#)
- [XSLTProcessor->transformToURI\(\)](#)
- [XSLTProcessor->transformToXML\(\)](#)

8.143.8 XSLTProcessor->getParameter()() : Récupère la valeur d'un paramètre

XSLTProcessor string **getParameter** (string namespaceURI , string localName)

Récupère un paramètre précédemment défini par la fonction [XSLTProcessor->setParameter\(\)](#) .

8.143.9 XSLTProcessor->hasExsltSupport()() : Indique si PHP utilise EXSLT

XSLTProcessor bool **hasExsltSupport** (void)

Cette méthode détermine si PHP a été compilé avec la bibliothèque [EXSLT](#) .

8.143.10 XSLTProcessor->importStylesheet()() : Importe une feuille de style

XSLTProcessor void **importStylesheet** (DOMDocument stylesheet)

Cette méthode importe une feuille de style dans le XSLTProcessor pour transformations.

8.143.11 XSLTProcessor->registerPHPFunctions()() : Active l'utilisation de PHP dans les feuilles de styles XSLT

XSLTProcessor void **registerPHPFunctions** (*mixed* restrict)

Cette méthode permet d'utiliser les fonctions PHP en tant que fonctions XSLT dans les feuilles de style XSL.

8.143.12 XSLTProcessor->removeParameter()() : Efface un paramètre

XSLTProcessor bool **removeParameter** (string namespaceURI , string localName)

Efface un paramètre, s'il est défini. Ceci fera que le processeur utilisera la valeur par défaut pour le paramètre comme défini dans la feuille de style.

8.143.13 XSLTProcessor->setParameter()() : Définit la valeur d'un paramètre

XSLTProcessor bool **setParameter** (string namespace , string name , string value)
XSLTProcessor bool **setParameter** (string namespace , array options)

Définit la valeur d'un ou plusieurs paramètres pour être utilisé dans une sous-séquence de transformation avec XSLTProcessor . Si le paramètre n'existe pas dans la feuille de style, il sera ignoré.

8.143.14 XSLTProcessor->transformToDoc()() : Transforme en un document DOM

XSLTProcessor DOMDocument **transformToDoc** (DOMNode doc)

Transforme le noeud source en un DOMDocument en y appliquant la feuille de style donnée par la méthode XSLTProcessor->importStylesheet() .

8.143.15 XSLTProcessor->transformToURI()() : Transforme en URI

XSLTProcessor int **transformToURI** (DOMDocument doc , string uri)

Transforme le noeud source en une URI en y appliquant la feuille de style donnée par la méthode XSLTProcessor->importStylesheet() .

8.143.16 XSLTProcessor->transformToXML()() : Transforme en XML

XSLTProcessor string **transformToXML** (DOMDocument doc)

Transforme le noeud source en une chaîne en y appliquant une feuille de style donnée par la méthode XSLTProcessor->importStylesheet() .

8.144 XSLT

8.144.1 Introduction

Cette extension PHP fourni une API indépendante pour les transformations XSLT. Actuellement, seul Sablotron est supporté. Le support d'autres bibliothèques comme Xalan ou libxslt est prévu.

XSLT (Extensible Stylesheet Language (XSL) Transformations) est un langage de transformation des documents XML en d'autres documents XML. C'est un standard défini par le consortium World Wide Web (W3C). Les informations sur le XSLT et ses technologies sont disponibles à <http://www.w3.org/TR/xslt> .

Note
Cette extension est différente de l'extension sablotron qui était distribuée dans les versions de PHP avant la 4.1. Actuellement, seule la nouvelle extension XSLT est supportée. Si vous souhaitez du support pour les anciennes extensions, posez vos questions sur les listes de diffusions PHP.
Note
Cette extension a été déplacée dans le module <u>PECL</u> et ne sera plus intégrée dans PHP à partir de PHP 5.0.0.
Note
Si vous avez besoin du support xslt avec PHP 5, vous pouvez utiliser l'extension <u>XSL</u> .

8.144.2 Pré-requis

Cette extension utilise Sablotron et expat , qui sont toutes les deux disponibles à <http://www.gingerall.com/> . Les sources comme les exécutables sont proposés.

8.144.3 Installation

Sous UNIX, lancez configure avec l'option --with-sablot ou --with-xslt-sablot . La bibliothèque Sablotron doit être installée là où le compilateur peut la trouver.

Assurez-vous d'avoir la même bibliothèque, compilée avec Sablotron que celle qui est fournie avec PHP. Les options de configuration : --with-expat-dir=DIR --with-iconv-dir=DIR sont disponibles pour vous aider à les spécifier correctement. Lorsque vous demandez de l'aide, mentionnez toujours ces directives, ainsi que si vous avez d'autres bibliothèques installées sur votre système. Naturellement, fournissez les numéros de version.

Attention
Assurez-vous que votre bibliothèque Sablotron soit correctement reliée à -lstdc++ . Dans le cas contraire, votre configuration échouera ou bien PHP refusera de fonctionner ou de se charger.
Note
Support de JavaScript E-XSLT
Si vous compilez Sablotron avec le support JavaScript, vous devez spécifier cette option : --with-sablot-js .
Note
Note aux utilisateurs Win32

Afin d'activer ce module dans l'environnement Windows, vous devez copier les bibliothèques **sablot.dll** depuis le dossier DLL de PHP/Win32 dans le dossier système SYSTEM32 de votre machine (par exemple : C:\WINNT\SYSTEM32 or C:\WINDOWS\SYSTEM32). Pour les versions <= à 4.2.0 , copiez sablot.dll et expat.dll dans le dossier système SYSTEM32. Pour les versions >= à 4.2.1, copiez sablot.dll , expat.dll et iconv.dll dans le dossier système SYSTEM32.

8.144.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.144.5 Types de ressources

Cette extension ne définit aucune ressource.

8.144.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

XSLT_OPT_SILENT (entier)

Ignore tous les rapports d'erreurs. C'est une option générique pour toutes les interfaces qui seront ajoutées plus tard.

XSLT_SABOPT_PARSE_PUBLIC_ENTITIES (entier)

Commande à Sablotron de faire l'analyse des entités publics. Par défaut, cette option est désactivée.

XSLT_SABOPT_DISABLE_ADDING_META (entier)

Ne pas ajouter le méta-tag "Content-Type" pour les générations HTML. La valeur par défaut est choisie lors de la compilation de Sablotron .

XSLT_SABOPT_DISABLE_STRIPPING (entier)

Arrête la suppression des espaces (pour les fichiers de données uniquement).

XSLT_SABOPT_IGNORE_DOC_NOT_FOUND (entier)

Considère les documents non résolus (la fonction document()) comme non-bloquant.

XSLT_SABOPT_FILES_TO_HANDLER (entier)

XSLT_ERR_UNSUPPORTED_SCHEME (entier)

Renvoie un code d'erreur pour les erreurs de gestionnaire de Scheme .

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [xslt_backend_info](#)
- [xslt_backend_name](#)
- [xslt_backend_version](#)
- [xslt_create](#)
- [xslt_erno](#)
- [xslt_error](#)
- [xslt_free](#)
- [xslt_getopt](#)
- [xslt_process](#)

- [xslt_set_base](#)
- [xslt_set_encoding](#)
- [xslt_set_error_handler](#)
- [xslt_set_log](#)
- [xslt_set_object](#)
- [xslt_set_sax_handler](#)
- [xslt_set_sax_handlers](#)
- [xslt_set_scheme_handler](#)
- [xslt_set_scheme_handlers](#)
- [xslt_setopt](#)

8.144.8 xslt_backend_name() : Retourne le nom du backend

string **xslt_backend_name** (void)

[xslt_backend_name](#) retourne toujours la chaîne Sablotron .

8.144.9 xslt_backend_version() : Retourne le numéro de version de Sablotron

string **xslt_backend_version** (void)

[xslt_backend_version](#) retourne le numéro de version de Sablotron s'il est disponible, et FALSE sinon.

8.144.10 xslt_create() : Crée un nouvel analyseur XSLT

resource **xslt_create** (void)

[xslt_create](#) retourne une ressource d'analyseur XSLT qui sera nécessaire pour les appels ultérieurs aux fonctions XSLT .

8.144.11 xslt_errno() : Retourne le numéro d'erreur XSLT courant

int **xslt_errno** (resource xh)

[xslt_errno](#) retourne le numéro de la dernière erreur, pour l'analyseur XSLT xh .

8.144.12 xslt_error() : Retourne un message d'erreur

string **xslt_error** (resource xh)

[xslt_errno](#) retourne le dernier message d'erreur, pour l'analyseur xh .

8.144.13 xslt_free() : Détruit un analyseur XSLT

```
void xslt_free ( resource xh )
```

xslt_free détruit l'analyseur XSLT xh .

8.144.14 xslt_getopt() : Récupère les options d'un processeur xsl donné

```
int xslt_getopt ( resource processor )
```

xslt_getopt retourne les options du processeur processor .

8.144.15 xslt_process() : Transforme des données XML avec XSLT

```
mixed xslt_process ( resource xh , string xmlcontainer , string xslcontainer , string resultcontainer , array arguments , array parameters )
```

xslt_process fait partie de la nouvelle vague de fonctions XSLT . Elle vous permet de réaliser des transformations XSLT en utilisant n'importe quel type de source : des conteneurs. Cela se fait grâce à l'utilisation de buffers d'arguments : un concept issu de Sablotron (actuellement, c'est le seul processeur XSLT qui le supporte). Les conteneurs sont par défaut des fichiers qui contiennent le document à traiter. Le conteneur de résultat est par défaut un fichier qui recevra le document traité. Si le conteneur de résultat n'est pas fourni (l'argument vaut alors NULL), le résultat sera retourné par la fonction.

Attention

Depuis PHP 4.0.6, cette fonction n'accepte plus de chaînes XML dans les paramètres xmlcontainer ou xslcontainer . Passer une chaîne contenant du XML dans un de ces paramètres engendra une erreur de segmentation dans les versions Sablotron supérieures ou égales à 0.95.

Les conteneurs peuvent aussi être modifiés via le tableau arguments .

Le type le plus simple de transformation réalisable avec xslt_process est la transformation d'un fichier XML avec un fichier XSLT , en plaçant le résultat dans un troisième fichier. Faire cela avec Sablotron est très facile...

Utilisation de xslt_process pour transformer un fichier XML avec un fichier XSL en un autre fichier XML

```
<?php
/* Allocation du processeur XSLT */
$xh = xslt_create();

/* Traitement du document */
if (xslt_process($xh, 'sample.xml', 'sample.xsl', 'result.xml')) {
    echo "Réussi. sample.xml a été transformé par sample.xsl en result.xml";
    echo ", result.xml contient ceci : \n<br />\n";
    echo "<pre>\n";
    readfile('result.xml');
    echo "</pre>\n";
} else {
    echo "Désolé, sample.xml n'a pu être transformé par sample.xsl en ";
    echo " result.xml. La raison est " . xslt_error($xh) . " et ";
    echo " le code d'erreur est " . xslt_errno($xh);
}
```

```
xslt_free($xh);

?>
```

Même si cette fonctionnalité est très pratique, il arrive souvent qu'en environnement web, vous vouliez être capable d'afficher directement votre résultat. Par conséquent, si vous omettez le troisième argument de `xslt_process`, ou que vous fournissez la valeur NULL à la place, la fonction va automatiquement retourner le résultat de la transformation XSLT, au lieu de l'écrire dans un fichier.

Utilisation de `xslt_process` pour transformer un fichier XML avec un fichier XSL et le placer dans une variable

```
<?php

/* Allocation d'un analyseur XSLT */
$xh = xslt_create();

/* Traitement du document */
$result = xslt_process($xh, 'sample.xml', 'sample.xsl');
if ($result) {
    echo "Réussi. sample.xml a été transformé par sample.xsl en result.xml";
    echo ", result.xml contient ceci : \n<br />\n";
    echo "<pre>\n";
    echo $result;
    echo "</pre>\n";
} else {
    echo "Désolé, sample.xml n'a pu être transformé par sample.xsl en ";
    echo " result.xml. La raison est " . xslt_error($xh) . " et ";
    echo " le code d'erreur est " . xslt_errno($xh);
}

xslt_free($xh);

?>
```

Les deux cas de transformations XSLT ci-dessus sont les cas les plus simples, quand on parle de transformation XSLT, et ce sont les cas les plus courants, mais il existe des situations où vous obtenez vos données XML et XSLT de sources externes comme des sockets ou des bases de données. Dans ces cas, vous avez déjà les données dans une variable, et en mode de production, écrire ces données dans des fichiers serait trop inefficace. C'est là où la syntaxe de buffers d'arguments de XSLT prend la relève. Au lieu de fichier en terme d'arguments XML et XSLT, vous pouvez passer des buffers d'arguments, qui sont alors substitués en valeur au cinquième argument de la fonction `xslt_process`. L'exemple suivant vous montre comment traiter du XML et du XSLT issus de variable, et de retrouver le résultat dans une troisième variable.

Utilisation de `xslt_process` pour transformer une variable XML avec une autre variable XSL et placer le résultat dans une variable

```
<?php

/* $xml et $xsl contiennent des données XML et XSL */

$args = array(
    '/_xml' => $xml,
    '/_xsl' => $xsl
);

/* Allocation d'un analyseur XSLT */
$xh = xslt_create();

/* Traitement du document */
$result = xslt_process($xh, 'arg:/_xml', 'arg:/_xsl', NULL, $args);
```

```

if ($result) {
    echo "Réussi. sample.xml a été transformé par sample.xsl en result.xml";
    echo ", result.xml contient ceci : \n<br />\n";
    echo "<pre>\n";
    echo $result;
    echo "</pre>\n";
}
else {
    echo "Désolé, sample.xml n'a pu être transformé par sample.xsl en ";
    echo " result.xml. La raison est " . xslt_error($xh) . " et ";
    echo " le code d'erreur est " . xslt_errno($xh);
}
xslt_free($xh);
?>

```

Enfin, le dernier argument de `xslt_process` représente un tableau de tous les paramètres que vous souhaitez passer au document XSLT . Ces paramètres sont accessibles depuis le fichier XSL en utilisant l'instruction `<xsl:param name="parameter_name">` . Les paramètres doivent être encodés au format UTF-8 et leur valeur sera interprétée comme une chaîne de caractères par l'analyseur Sablotron . En d'autres termes, vous ne pouvez pas passer des groupes de noeuds (node-sets) comme paramètre au document XSLT .

Passer des variables PHP à un fichier XSL

```

<?php

// XML string
$xml = '
<para>
  change me
</para>';

// XSL string
$xsl = '
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="ISO-8859-1" indent="no"
  omit-xml-declaration="yes" media-type="text/html"/>
<xsl:param name="myvar"/>
<xsl:param name="mynode"/>
<xsl:template match="/">
Ma variable PHP : <xsl:value-of select="$myvar"/><br />
Mon node set : <xsl:value-of select="$mynode"/>
</xsl:template>
</xsl:stylesheet>';

$xh = xslt_create();

// le deuxième paramètre sera interprété comme une chaîne
$parameters = array (
  'myvar' => 'test',
  'mynode' => '<foo>bar</foo>'
);

$args = array (
  '/_xml' => $xml,
  '/_xsl' => $xsl
);

echo xslt_process($xh, 'arg:/_xml', 'arg:/_xsl', NULL, $args, $parameters);

?>

```

L'exemple ci-dessus va afficher :

Ma variable PHP : test Mon node set : <foo>bar</foo>

Note

Notez que vous devez préfixer le chemin par file:// si vous utilisez Windows.
--

8.144.16 `xslt_set_base()` : Configure l'URI de base de toutes les transformations XSLT

```
void xslt_set_base ( resource xh , string uri )
```

`xslt_set_base` configure l'URI de base de toutes les transformations XSLT . L'URI est utilisée avec les instructions Xpath pour résoudre les commandes `document()`, et les autres commandes qui ont des accès externes. Il est également utilisé pour résoudre les URIs pour les éléments `<xsl:include>` et `<xsl:import>`.

A partir de PHP 4.3, la valeur par défaut de l'URI est le répertoire courant d'exécution du script. En effet, c'est la nom du répertoire contenu dans la constante `__FILE__` . Dans les versions antérieures à 4.3, la valeur par défaut de l'URI était moins prévisible.

Note

Notez que vous devez préfixer le chemin par file:// si vous utilisez Windows.
--

8.144.17 `xslt_set_encoding()` : Configure le jeu de caractères pour l'analyse des documents XML

```
void xslt_set_encoding ( resource xh , string encoding )
```

`xslt_set_encoding` configure le jeu de caractères pour l'analyse des documents XML . Lorsque vous utilisez Sablotron , cette option est uniquement disponible lorsque vous compilez Sablotron avec le support des jeux de caractères.

8.144.18 `xslt_set_error_handler()` : Configure le gestionnaire d'erreurs du processeur XSLT

```
void xslt_set_error_handler ( resource xh , mixed handler )
```

`xslt_set_error_handler` configure le gestionnaire d'erreur du processeur XSLT identifié par `xh` . Le gestionnaire sera appelé à chaque fois qu'une erreur survient durant les transformations XSLT (pour les notices aussi).

La fonction utilisateur doit accepter quatre paramètres : le processeur XSLT, le niveau d'erreur, le code erreur et un tableau de messages. La fonction peut être définie comme : **error_handler**

```
resource xh ( int error_level , int error_code , array messages )
```

8.144.19 `xslt_set_log()` : Configure le fichier d'historique pour les messages XSLT

```
void xslt_set_log ( resource xh , mixed log )
```

xh	Une ressource d'analyseur XSLT .
log	Ce paramètre doit être un booléen, qui active ou non l'enregistrement dans le fichier d'historique.

xslt_set_log vous permet de configurer le fichier dans lequel vous voulez enregistrer les messages XSLT . Les messages XSLT sont différents des messages d'erreurs, car les messages ne sont pas des messages d'erreurs, mais plutôt des descriptions de l'état de l'analyseur. Ils sont très pratiques pour déboguer du code XSLT .

Par défaut, l'enregistrement dans l'historique est désactivé. Afin de l'activer, vous devez appeler xslt_set_log avec le booléen qui active l'enregistrement, puis, si vous le voulez, indiquer un fichier de débogage. Il faut alors passer un nom de fichier.

Note

Notez que vous devez préfixer le chemin par file:// si vous utilisez Windows.
--

8.144.20 xslt_set_object() : Définie l'objet dans lequel doivent être résolues les fonctions de callback

bool **xslt_set_object** (resource processor , object obj)

xslt_set_object autorise l'utilisation du paramètre processor dans un objet object et donc de résoudre toutes les fonctions de callback dans celui-ci.

Les fonctions de callback peuvent être déclarées avec les fonctions xml_set_sax_handlers , xslt_set_scheme_handlers ou xslt_set_error_handler et sont supposées comme des méthodes de l'objet object .

8.144.21 xslt_set_sax_handler() : Modifie les gestionnaires SAX de l'analyseur XSLT

void **xslt_set_sax_handler** (resource xh , array handlers)

xslt_set_sax_handler remplace les gestionnaires SAX de l'analyseur XSLT xh par handlers . Les gestionnaires SAX sont représentés par un tableau à deux dimensions, avec le format suivant (les éléments du premier niveau sont optionnels) :

Format des gestionnaires avec <u>xslt_set_sax_handler</u>

<pre>array([document] => array(start document handler, end document handler), [element] => array(start element handler, end element handler), [namespace] => array(start namespace handler,</pre>
--

```

        end namespace handler
    ),
    [comment] => comment handler,
    [pi] => processing instruction handler,
    [character] => character data handler
)

```

8.144.22 `xslt_set_sax_handlers()` : Configure les gestionnaires SAX qui seront appelés pour gérer les documents XML

void **xslt_set_sax_handlers** (resource processor , array handlers)

`xslt_set_sax_handlers` enregistre le gestionnaire SAX handlers pour le document, en lui donnant une ressource XSLT processor .

Le paramètre handlers doit être un tableau qui suit ce format :

```

<?php
$handlers = array(

    "document" => array(
        "start_doc",
        "end_doc"),

    "element" => array(
        "start_element",
        "end_element"),

    "namespace" => array(
        "start_namespace",
        "end_namespace"),

    "comment"    => "comment",

    "pi"         => "pi",

    "character"  => "characters"

);
?>

```

Où les fonctions suivent la syntaxe décrite dans le schéma du gestionnaire de fonctions.

Note

Le tableau fournit n'a pas besoin de contenir tous les éléments différents du gestionnaire SAX (bien qu'il le peut), mais il doit uniquement être conforme au format "gestionnaire" => "fonction" décrite ci-dessus.

Chacune des fonctions du gestionnaire SAX correspondent au format suivant :

- **start_doc** resource processor ()
- **end_doc** resource processor ()
- **start_element** resource processor (string name , array attributes)
- **end_element** resource processor (string name)
- **start_namespace** resource processor (string prefix , string uri)
- **end_namespace** resource processor (string prefix)
- **comment** resource processor (string contents)
- **pi** resource processor (string target , string contents)
- **characters** resource processor (string contents)

Utiliser la fonction `xslt_set_sax_handlers` n'est pas vraiment différent qu'exécuter un analyseur SAX comme `xml_parse` sur un résultat de la transformation `xslt_process` .

8.144.23 `xslt_set_scheme_handler()` : Configure les gestionnaires de Scheme du processeur XSLT

void `xslt_set_scheme_handler` (resource xh , array handlers)

`xslt_set_scheme_handler` configure les gestionnaires de Scheme, pour le processeur XSLT identifié par sa ressource xh . Les gestionnaires Scheme doivent être fournis sous forme de tableau (tous les éléments sont optionnels) :

Configuration des gestionnaires de Scheme pour XSLT

```
array(
[get_all] => get all handler,
[open] => open handler,
[get] => get handler,
[put] => put handler,
[close] => close handler
)
```

8.144.24 `xslt_set_scheme_handlers()` : Configure un gestionnaire de Scheme pour un processeur XSLT

void `xslt_set_scheme_handlers` (resource processor , array handlers)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.144.25 `xslt_setopt()` : Définit les options d'un processeur xsl donné

mixed `xslt_setopt` (resource processor , int newmask)

`xslt_setopt` définit les options newmask pour le processeur processor . Retourne le nombre de masques précédent possible, TRUE sinon.

Le paramètre newmask est un masque construit avec les constantes suivantes :

- `XSLT_SABOPT_PARSE_PUBLIC_ENTITIES` - Demande au processeur d'analyser les entrées publiques. Par défaut, cette constante est désactivée.
- `XSLT_SABOPT_DISABLE_ADDING_META` - Permet de ne pas ajouter l'en-tête méta " Content-Type " lors de l'affichage HTML. La valeur par défaut est définie lors de la compilation du processeur.
- `XSLT_SABOPT_DISABLE_STRIPPING` - Supprime les espaces (uniquement pour les fichiers de données).
- `XSLT_SABOPT_IGNORE_DOC_NOT_FOUND` - Considère les documents non résolus (la fonction `document()`) comme existant.

8.145 YAZ

8.145.1 Introduction

Cette extension offre à PHP l'interface avec les produits YAZ , qui implémentent le protocole Z39.50 . Avec cette extension, vous pouvez facilement implémenter un client Z39.50 qui analyse ou scanne des serveurs Z39.50 en parallèle.

Le module masque l'essentiel de la complexité de Z39.50, ce qui le rend très facile à utiliser. Il supporte les connexions persistantes de manière similaire à celles supportées par les serveurs SQL : cela signifie qu'une connexion est partagée entre plusieurs scripts PHP, ce qui évite les opérations de connexions.

YAZ est disponible sur <http://www.indexdata.dk/yaz/> . Vous pouvez trouver des informations, des scripts d'exemples, etc. pour cette extension sur <http://www.indexdata.dk/phpyaz/> .

Note

Cette extension a été déplacée dans le module PECL et ne sera plus intégrée dans PHP à partir de PHP 5.0.0.

8.145.2 Installation

Obtenez YAZ (ANSI/NISO Z39.50 support) et installez-le. YAZ peut être récupéré en source ou dans des paquetages déjà compilés à partir des archives YAZ . Les systèmes tels que Debian GNU/Linux, Suse Linux, FreeBSD ont aussi YAZ fourni dans leur distribution.

Pour les versions de PHP 4, l'extension YAZ y est intégré (mais pas YAZ soi-même). Compiler PHP avec vos modules favoris et ajoutez l'option `--with-yaz[=DIR]` .

Compilation de YAZ pour PHP 4 sur Unix

```
gunzip -c php-4.4.X.tar.gz|tar xf -
gunzip -c yaz-2.1.8.tar.gz|tar xf -
cd yaz-2.1.8
./configure --prefix=/usr
make
sudo make install
cd ../php-4.4.X.
./configure --with-yaz=/usr/bin
make
sudo make install
```

Pour PHP 5, l'extension YAZ est dans PECL et est installée en tant qu'objet/dll partagé. Si vous avez peur d'installé, la plus facile méthode pour télécharger, configurer et installer l'extension YAZ est d'utiliser la commande `pear` .

Installation de YAZ PECL sur Linux Gnu/Debian

L'exemple suivant installe YAZ et YAZ PECL sur un système Debian GNU/Linux :

```
sudo apt-get install libyaz-dev
sudo pear install yaz
```

Sur Windows, `php_yaz.dll` dépend de `yaz.dll` . Le fichier `yaz.dll` est une partie du fichier ZIP Win32 provenant du site de PHP. Il s'agit aussi d'une partie de l'installation de Windows YAZ disponible sur YAZ WIN32 area .

Attention

Le fichier zip Win32 de PHP 5.0.5 inclue une version trop vieille de yaz.dll (version 1.9.1 < version 2.0.13 requise). Si tel est le cas, utilisez, yaz.dll à partir d'une nouvelle [installation YAZ WIN32](#) .

Sur windows, n'oubliez pas d'ajouter le répertoire de PHP à votre variable d'environnement PATH pour que le fichier yaz.dll puisse être trouvé par votre système.

Si vous utilisez YAZ sous forme d'extension partagée, ajoutez (ou décommentez) la ligne suivante dans le fichier php.ini sous Unix:

```
extension=php_yaz.so
```

Et sous Windows :

```
extension=php_yaz.dll
```

Attention

L'extension [IMAP](#) ne peut être utilisée simultanément avec l'extension [recode](#) , l'extension [YAZ](#) ou l'extension [Cyrus](#) . Ceci est dû au fait que ces bibliothèques utilisent un symbole interne commun.

Note

Le problème ci-dessus est résolu en version 2.0 de YAZ.

8.145.3 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

Nom	Par défaut	Modifiable	Historique
yaz.max_links	"100"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
yaz.log_file	NULL	PHP_INI_ALL	Disponible depuis PHP 4.3.0.

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à [Directives du php.ini](#) .

8.145.4 Types de ressources

Cette extension ne définit aucune ressource.

8.145.5 Constantes pré-définies

Cette extension ne définit aucune constante.

8.145.6 Exemples

PHP/YAZ conserve les connexions aux serveurs. Une ressource de connexion représente une connexion particulière.

Le script ci-dessous montre comment effectuer une recherche parallèle. Lorsqu'il est appelé sans paramètres, ce script affiche la requête. Sinon, il effectue la recherche sur les serveurs donnés par le tableau host .

Recherche parallèle utilisant YAZ

```
<?php
$host=$_REQUEST[host];
$query=$_REQUEST[query];
$num_hosts = count($host);
```

```

if (empty($query) || count($host) == 0) {
    echo '<form method="get">
    <input type="checkbox"
    name="host[]" value="bagel.indexdata.dk/gils" />
        GILS test
    <input type="checkbox"
    name="host[]" value="localhost:9999/Default" />
        local test
    <input type="checkbox" checked="checked"
    name="host[]" value="z3950.loc.gov:7090/voyager" />
        Library of Congress
    <br />
    RPN Query:
    <input type="text" size="30" name="query" />
    <input type="submit" name="action" value="Search" />
    </form>
    ';
} else {
    echo 'Vous avez recherché '.htmlspecialchars($query).'<br />';
    for ($i = 0; $i < $num_hosts; $i++) {
        $id[] = yaz_connect($host[$i]);
        yaz_syntax($id[$i], "usmarc");
        yaz_range($id[$i], 1, 10);
        yaz_search($id[$i], "rpn", $query);
    }
    yaz_wait();
    for ($i = 0; $i < $num_hosts; $i++) {
        echo '<hr />' . $host[$i] . ':';
        $error = yaz_error($id[$i]);
        if (!empty($error)) {
            echo "Erreur: $error";
        } else {
            $hits = yaz_hits($id[$i]);
            echo "Nombre de résultats : $hits";
        }
        echo '<dl>';
        for ($p = 1; $p <= 10; $p++) {
            $rec = yaz_record($id[$i], $p, "string");
            if (empty($rec)) continue;
            echo "<dt><strong>$p</strong></dt><dd>";
            echo nl2br($rec);
            echo "</dd>";
        }
        echo '</dl>';
    }
}
?>

```

- [Introduction](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [yaz_addinfo](#)
- [yaz_ccl_conf](#)
- [yaz_ccl_parse](#)
- [yaz_close](#)
- [yaz_connect](#)
- [yaz_database](#)
- [yaz_element](#)
- [yaz_errno](#)

- [yaz_error](#)
- [yaz_es_result](#)
- [yaz_get_option](#)
- [yaz_hits](#)
- [yaz_itemorder](#)
- [yaz_present](#)
- [yaz_range](#)
- [yaz_record](#)
- [yaz_scan_result](#)
- [yaz_scan](#)
- [yaz_schema](#)
- [yaz_search](#)
- [yaz_set_option](#)
- [yaz_sort](#)
- [yaz_syntax](#)
- [yaz_wait](#)

8.145.8 [yaz_ccl_conf\(\)](#) : Configure l'analyseur CCL

void [yaz_ccl_conf](#) (resource id , array config)

Cette fonction configure l'analyseur CCL de requête pour un serveur avec les définitions de points d'accès (CCL qualifieurs) et leur équivalent en RPN.

Pour faire correspondre une requête CCL spécifique à RPN, utilisez la fonction [yaz_ccl_parse](#) .

8.145.9 [yaz_ccl_parse\(\)](#) : Appelle l'analyseur CCL

bool [yaz_ccl_parse](#) (resource id , string query , array result)

Cette fonction appelle l'analyseur CCL. Il convertit une requête CCL FIND en une requête RPN qui peut être passée à [yaz_search](#) pour effectuer une recherche.

Pour définir un champ CCL valide, utilisez la fonction [yaz_ccl_conf](#) avant d'utiliser cette fonction.

8.145.10 [yaz_close\(\)](#) : Ferme une connexion YAZ

bool [yaz_close](#) (resource id)

Ferme la connexion donnée par le paramètre id .

Note

Cette fonction fermera seulement une connexion non persistante ouverte en mettant l'option persistant à FALSE avec yaz_connect .
--

8.145.11 [yaz_connect\(\)](#) : Prépare une connexion à un serveur Z39.50

mixed [yaz_connect](#) (string zurl , mixed options)

Cette fonction retourne une ressource de connexion en cas de succès et 0 en cas d'échec.

`yaz_connect` prépare une connexion à un serveur Z39.50. Cette fonction n'est pas bloquante, et ne tente pas d'établir une connexion. En fait, elle ne fait que préparer la connexion pour exécution ultérieure par `yaz_wait`.

Note

Le proxy <code>YAZ</code> est un proxy Z39.50 librement disponible.

8.145.12 `yaz_database()` : Spécifie la base d'une session YAZ

bool `yaz_database` (resource id , string databases)

Cette fonction vous permet de changer de base de données à l'intérieur de la session en spécifiant une ou plusieurs bases de données à utiliser dans la recherche, récupération de données, etc. Écrase les bases de données spécifiées dans l'appel de `yaz_connect`.

8.145.13 `yaz_element()` : Spécifie le type d'éléments YAZ à lire

bool `yaz_element` (resource id , string elementset)

Cette fonction spécifie le type d'élément à lire.

Appelez cette fonction avant `yaz_search` ou `yaz_present` pour spécifier un type d'élément pour les enregistrements à récupérer.

Note

Si cette fonction semble sans effet, voyez la description de l'option piggybacking dans <code>yaz_connect</code> .
--

8.145.14 `yaz_errno()` : Retourne le numéro d'erreur

int `yaz_errno` (resource id)

Retourne un numéro d'erreur pour le serveur (dernière requête) identifié par id .

`yaz_errno` devrait être appelée après chaque requête pour chaque serveur - (après le retour de `yaz_wait`) pour déterminer le succès ou l'échec de la dernière opération (c'est-à-dire la recherche).

8.145.15 `yaz_error()` : Retourne une description de l'erreur

string `yaz_error` (resource id)

`yaz_error` retourne un message d'erreur anglais correspondant au dernier numéro d'erreur retourné par `yaz_errno`.

8.145.16 `yaz_es_result()` : Inspecte le résultat Extended Services

array `yaz_es_result` (resource id)

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

8.145.17 yaz_get_option() : Retourne la valeur d'une option de connexion YAZ

string **yaz_get_option** (resource id , string name)

Retourne la valeur de l'option spécifiée par name .

8.145.18 yaz_hits() : Retourne le nombre de résultats de la dernière recherche

int **yaz_hits** (resource id , array searchresult)

yaz_hits retourne le nombre de résultats de la dernière recherche.

8.145.19 yaz_itemorder() : Prépare une requête Z39.50 Item Order avec le package ILL-Request

void **yaz_itemorder** (resource id , array args)

Cette fonction prépare une requête de type "Extended Services" en utilisant le "Profile" avec "Use of Z39.50 Item Order Extended Service to Transport ILL (Profile/1)". Reportez-vous [ici](#) ou aux [spécifications](#) .

8.145.20 yaz_present() : Prépare à la lecture (Z39.50 present)

bool **yaz_present** (resource id)

Cette fonction prépare PHP à la lecture des résultats, après une recherche.

La fonction yaz_range doit être appelée avant celle-ci pour spécifier la plage de résultats à lire.

8.145.21 yaz_range() : Spécifie le nombre maximal de résultats à lire

void **yaz_range** (resource id , int start , int number)

Spécifie le nombre maximal de résultat à lire.

Cette fonction doit être appelé avant la fonction yaz_search ou yaz_present .

8.145.22 yaz_record() : Retourne un résultat

string **yaz_record** (resource id , int pos , string type)

La fonction yaz_record retourne un enregistrement dans le jeu de résultats courant à la position spécifiée par le paramètre pos .

8.145.23 `yaz_scan_result()` : Retourne le résultat d'un scan

`array yaz_scan_result (resource id , array result)`

`yaz_scan_result` retourne un tableau contenant les termes reçus de l'hôte, lors de la dernière requête de scan.

8.145.24 `yaz_scan()` : Prépare un scan

`void yaz_scan (resource id , string type , string startterm , array flags)`

Cette fonction prépare une requête "Z39.50 Scan Request" sur la connexion spécifiée.

Pour actuellement transférer la requête "Scan Request" au serveur et recevoir le "Scan Response", `yaz_wait` doit être appelée. Après la fin de `yaz_wait`, appelez `yaz_error` et `yaz_scan_result` pour gérer la réponse.

8.145.25 `yaz_schema()` : Spécifie le schéma de lecture

`void yaz_schema (resource id , string schema)`

`yaz_schema` spécifie le schéma de lecture.

cette fonction devrait être appelée avant `yaz_search` ou `yaz_present`.

8.145.26 `yaz_search()` : Prépare une recherche

`bool yaz_search (resource id , string type , string query)`

`yaz_search` prépare une recherche sur la connexion donnée

Comme pour `yaz_connect`, cette fonction n'est pas bloquante et ne fait que préparer la recherche pour exécution ultérieure lorsque `yaz_wait` est appelée.

8.145.27 `yaz_set_option()` : Modifie une ou plusieurs options de connexion YAZ

`void yaz_set_option (resource id , string name , string value) void yaz_set_option (resource id , array options)`

Modifie une ou plusieurs options sur la connexion donnée.

8.145.28 `yaz_sort()` : Configure les critères de tri

`void yaz_sort (resource id , string criteria)`

Cette fonction configure les critères de tri et active le tri Z39.50.

Appelez cette fonction **avant** yaz_search . Si elle est utilisée conjointement avec yaz_search , une commande Z39.50 Sort sera envoyée après chaque retour de recherche et avant que les résultats ne soient lus avec Z39.50 Present (yaz_present).

8.145.29 yaz_syntax() : Spécifie la syntaxe de lecture des lignes

void **yaz_syntax** (resource id , string syntax)

yaz_syntax spécifie la syntaxe de lecture des lignes.

Cette fonction devrait être appelée avant yaz_search ou yaz_present .

8.145.30 yaz_wait() : Attend l'exécution d'une requête

mixed **yaz_wait** (array options)

Cette fonction exécute une activité réseau (bloquée) pour les demandes en attente qui ont été préparées par les fonctions yaz_connect yaz_search , yaz_present , yaz_scan et yaz_itemorder

yaz_wait se termine lorsque tous les hôtes ont terminé leurs requêtes (éventuellement en cas d'erreur).

8.146 Accès aux fichiers Zip (Lecture seule)

8.146.1 Introduction

Ce module permet l'accès transparent aux archives compressées par la méthode ZIP : on peut y lire les fichiers et les dossiers qui y sont contenus.

8.146.2 Pré-requis

Ce module utilise les fonctions de la bibliothèque ZZIPLib , écrite par Guido Draheim. Vous aurez besoin de ZZIPLib version $\geq 0.10.6$.

Notez que ZZIPLib fournit une partie de l'implémentation complète de l'algorithme de compression ZIP, et peut uniquement lire des archives ZIP. Un utilitaire ZIP normal est nécessaire pour créer l'archive elle-même.

8.146.3 Installation

Cette extension PECL n'est pas intégrée à PHP. Des informations comme les notes sur les nouvelles versions, les téléchargements, les sources des fichiers, les informations concernant les mainteneurs ainsi qu'un CHANGELOG, peuvent être trouvées ici : <http://pecl.php.net/package/zip> .

En PHP 4, les sources de cette extension PECL peuvent être trouvées dans le dossier ext/ avec les sources de PHP ou sur le lien PECL ci-dessous. Si vous voulez utiliser ces fonctions, vous devez compiler PHP avec le support zip en utilisant l'option de configuration `--with-zip[=DIR]` .

Les utilisateurs de Windows doivent activer la bibliothèque php_zip.dll dans le php.ini pour pouvoir utiliser ces fonctions. En PHP 4, cette bibliothèque DLL se trouve dans le dossier extensions/ avec les binaires PHP pour Windows téléchargées. Vous pouvez télécharger ces bibliothèques DLL pour ces extensions PECL depuis la page de téléchargement de PHP sous sur la page <http://snaps.php.net/> .

Note

Le support Zip dans les versions antérieures à PHP 4.1.0 est expérimental. Cette section reflète l'extension Zip comme elle existe dans les version 4.1.0 et suivantes de PHP.
--

8.146.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

8.146.5 Types de ressources

Cette extension ne définit aucune ressource.

8.146.6 Constantes pré-définies

Cette extension ne définit aucune constante.

8.146.7 Exemples

Cet exemple ouvre un fichier ZIP, lit chaque fichier de l'archive, et en affiche le contenu. Le fichier d'archive test2.zip utilisé dans cet exemple est fourni dans la distribution source de ZZIPLib.

Exemple d'utilisation de ZIP

```
<?php
$zip = zip_open("/tmp/test2.zip");

if ($zip) {

    while ($zip_entry = zip_read($zip)) {
        echo "Nom : " . zip_entry_name($zip_entry) . "\n";
        echo "Taille réelle du fichier : " . zip_entry_filesize($zip_entry) . "\n";
        echo "Taille compressée du fichier : " . zip_entry_compressedsize($zip_entry) . "\n";
        echo "Méthode de compression : " . zip_entry_compressionmethod($zip_entry) . "\n";

        if (zip_entry_open($zip, $zip_entry, "r")) {
            echo "Contenu du fichier : \n";
            $buf = zip_entry_read($zip_entry, zip_entry_filesize($zip_entry));
            echo "$buf\n";

            zip_entry_close($zip_entry);
        }
        echo "\n";
    }

    zip_close($zip);
}
?>
```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [zip_close](#)
- [zip_entry_close](#)
- [zip_entry_compressedsize](#)
- [zip_entry_compressionmethod](#)
- [zip_entry_filesize](#)
- [zip_entry_name](#)
- [zip_entry_open](#)
- [zip_entry_read](#)
- [zip_open](#)
- [zip_read](#)

8.146.9 zip_entry_close() : Ferme un dossier d'archive

void **zip_entry_close** (resource zip_entry)

zip_entry_close ferme un dossier d'archive donné.

8.146.10 zip_entry_compressedsize() : Lit la taille compressée d'un dossier d'archives

int **zip_entry_compressedsize** (resource zip_entry)

zip_entry_compressedsize retourne la taille compressée d'un dossier d'archives donné.

8.146.11 zip_entry_compressionmethod() : Lit la méthode de compression utilisée sur un dossier d'archives

string **zip_entry_compressionmethod** (resource zip_entry)

zip_entry_compressionmethod retourne la méthode de compression utilisée sur le dossier d'archives spécifié par zip_entry .

8.146.12 zip_entry_filesize() : Lit la taille décompressée d'un dossier d'archives

int **zip_entry_filesize** (resource zip_entry)

zip_entry_filesize retourne la taille décompressée d'un dossier d'archives donné.

8.146.13 zip_entry_name() : Lit le nom d'un dossier d'archives

string **zip_entry_name** (resource zip_entry)

zip_entry_name retourne le nom d'un dossier d'archives donné.

8.146.14 zip_entry_open() : Ouvre un dossier d'archives en lecture

bool **zip_entry_open** (resource zip , resource zip_entry , string mode)

zip_entry_open ouvre un dossier dans un fichier ZIP pour lecture.

8.146.15 zip_entry_read() : Lit le contenu d'un fichier dans un dossier

string **zip_entry_read** (resource zip_entry , int length)

zip_entry_read lit dans un dossier d'archives ouvert.

8.146.16 zip_open() : Ouvre une archive ZIP

resource **zip_open** (string filename)

zip_open ouvre une nouvelle archive ZIP pour lecture.

8.146.17 zip_read() : Lit la prochaine entrée dans une archive ZIP

resource **zip_read** (resource zip)

zip_read lit la prochaine entrée dans une archive ZIP.

8.147 Compression Zlib

8.147.1 Introduction

Ce module vous permet de lire et écrire de manière transparente des fichiers compressés gzip (.gz), avec la plupart des fonctions fichier qui fonctionnent avec les fichiers compressés en gzip (et les fichiers décompressés aussi, mais pas avec les sockets).

Note

La version 4.0.4 a introduit la version étendue de fopen pour les fichiers compressés avec GZ. Vous pouvez utiliser le préfixe spécial zlib: dans les URL pour accéder directement et de manière transparente aux fichiers compressés par GZ, avec toutes les fonctions f*() habituelles. Cette fonctionnalité requiert une bibliothèque C qui fournit la fonction fopencookie() . A ma connaissance, la bibliothèque GNU libc est la seule à fournir cette fonctionnalité.

En version 4.3.0, zlib: a été changé pour compress.zlib:// pour éviter les ambiguïtés avec les fichiers contenant les caractères ':'. La fonction fopencookie() n'est plus nécessaire. Plus d'informations sont disponibles dans la section à propos des Flux compressés .

8.147.2 Pré-requis

Ce module utilise les fonctions de la bibliothèque zlib de Jean-loup Gailly et Mark Adler. Vous devez utiliser zlib version >= 1.0.9 avec ce module.

8.147.3 Installation

Le support de Zlib dans PHP n'est pas activé par défaut. Vous devez compiler PHP avec l'option --with-zlib[=DIR] .

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de bibliothèque supplémentaire pour disposer de ces fonctions.

Note

Le support intégré de zlib est disponible depuis PHP 4.3.0.

8.147.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

L'extension zlib offre l'option de compresser de manière transparente les pages PHP à la volée, si le navigateur du visiteur le supporte. Voici donc les trois options à utiliser dans le php.ini .

Nom	Par défaut	Modifiable	Historique
zlib.output_compression	"0"	PHP_INI_ALL	Disponible depuis PHP 4.0.5.
zlib.output_compression_level	"-1"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
zlib.output_handler	""	PHP_INI_ALL	Disponible depuis PHP 4.3.0.

Pour plus de détails sur les constantes PHP_INI_*, reportez-vous à Directives du php.ini .

Voici un éclaircissement sur l'utilisation des directives de configuration.

`zlib.output_compression` booléen / entier

Active ou pas la compression transparente des pages. Si cette option est mise à "On" dans `php.ini` ou dans la configuration Apache, les pages sont compressées si le navigateur envoie un en-tête "Accept-Encoding: gzip" ou "deflate". Les en-têtes "Content-Encoding: gzip" (respectivement "deflate") et "Vary: Accept-Encoding" sont ajoutés dans la page envoyée au navigateur. En fonctionnement, il peut être défini uniquement avant tout affichage.

Cette option accepte aussi des valeurs entières au lieu des booléens, "On"/"Off", ce qui vous permet de configurer la taille du tampon de sortie (par défaut, il vaut 4ko).

Note

<code>output_handler</code> doit être laissée à vide si cette option est activée. Sinon, vous devez utiliser <code>zlib.output_handler</code> .

`zlib.output_compression_level` entier

Niveau de compression utilisé pour la compression de sortie.

`zlib.output_handler` chaîne de caractères

Vous ne pouvez pas spécifier de gestionnaire de sortie supplémentaire si `zlib.output_compression` est activée. Cette configuration est la même que `output_handler` mais dans un ordre différent.

8.147.5 Types de ressources

Cette extension ne définit aucune ressource.

8.147.6 Constantes pré-définies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

`FORCE_GZIP` (entier)

`FORCE_DEFLATE` (entier)

8.147.7 Exemples

Cet exemple ouvre un fichier temporaire, écrit une page de test dedans, puis affiche le contenu du fichier deux fois.

Exemple de l'extension Zlib

<pre><?php \$filename = tempnam('/tmp', 'zlibtest') . '.gz'; echo "<html>\n<head></head>\n<body>\n<pre>\n"; \$s = "C'est juste un test, test, test, test, test, test, test!\n"; // ouvre le fichier en écriture avec la compression maximale \$zp = gzopen(\$filename, "w9"); // écrit une chaîne dans le fichier gzwrite(\$zp, \$s); // ferme le fichier</pre>

```

gzclose($zp);

// ouvre le fichier en lecture
$zp = gzopen($filename, "r");

// lit 3 caractères
echo gzread($zp, 3);

// affiche le reste du fichier, puis le ferme
gzpassthru($zp);
gzclose($zp);

echo "\n";

// ouvre le fichier et affiche le fichier une seconde fois
if (readgzfile($filename) != strlen($s)) {
    echo "Erreur avec les fonctions zlib!";
}
unlink($filename);
echo "</pre>\n</body>\n</html>\n";

?>

```

- [Introduction](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration à l'exécution](#)
- [Types de ressources](#)
- [Constantes pré-définies](#)
- [Exemples](#)
- [gzclose](#)
- [gzcompress](#)
- [gzdeflate](#)
- [gzencode](#)
- [gzeof](#)
- [gzfile](#)
- [gzgetc](#)
- [gzgets](#)
- [gzgetss](#)
- [gzinflate](#)
- [gzopen](#)
- [gzpassthru](#)
- [gzputs](#)
- [gzread](#)
- [gzrewind](#)
- [gzseek](#)
- [gztell](#)
- [gzuncompress](#)
- [gzwrite](#)
- [readgzfile](#)
- [zlib_get_coding_type](#)

8.147.9 gzcompress() : Comprime une chaîne

string **gzcompress** (string *data* , int *level*)

[gzcompress](#) compresse la chaîne donnée en utilisant le format de données ZLIB .

Pour plus de détails sur l'algorithme, lisez le document " [ZLIB Compressed Data Format Specification version 3.3](#) " (RFC 1950).

Note

Ce n'est pas la même chose que la compression gzip, qui inclut quelques en-têtes de données. Voir gzencode pour la compression gzip.

8.147.10 gzdeflate() : Comprime une chaîne

string **gzdeflate** (string data , int level)

[gzdeflate](#) compresse la chaîne donnée en utilisant le format de données DEFLATE .

Pour plus de détails concernant l'algorithme de compression DEFLATE, lisez le document " [DEFLATE Compressed Data Format Specification version 1.3](#) " (RFC 1951).

8.147.11 gzencode() : Crée une chaîne compressée gzip

string **gzencode** (string data , int level , int encoding_mode)

[gzencode](#) retourne la version compressée de la chaîne data compatible avec la sortie du programme gzip .

Pour plus de détails sur le format de fichier GZIP, lisez le document : [GZIP file format specification version 4.3](#) (RFC 1952).

8.147.12 gzeof() : Indique si la fin d'un fichier compressé est atteinte

int **gzeof** (resource zp)

[gzeof](#) indique si la fin d'un fichier compressé est atteinte.

8.147.13 gzfile() : Lit la totalité d'un fichier compressé

array **gzfile** (string filename , int use_include_path)

[gzfile](#) est identique à la fonction [readgzfile](#) , excepté qu'elle retourne le fichier dans un tableau.

8.147.14 gzgetc() : Lit un caractère dans un fichier compressé

string **gzgetc** (resource zp)

[gzgetc](#) retourne une chaîne contenant un seul caractère (décompressé) lu depuis le pointeur de fichier gz.

8.147.15 gzgets() : Lit une ligne dans un fichier compressé

string **gzgets** (resource *zp* , int *length*)

gzgets retourne une chaîne non compressée, de taille maximale *length* - 1 octets, lue dans le fichier représenté par *zp* . La lecture s'arrête lorsque *length* - 1 octets ont été lus, ou que la fonction a rencontré une nouvelle ligne ou la fin du fichier (le premier des trois qui survient).

8.147.16 gzgetss() : Lit une ligne dans un fichier compressé, et supprime les balises HTML

string **gzgetss** (resource *zp* , int *length* , string *allowable_tags*)

gzgetss est identique à la fonction gzgets , excepté que gzgetss tente de supprimer toutes les balises HTML et PHP du texte lu.

8.147.17 gzinflate() : Décompresse une chaîne

string **gzinflate** (string *data* , int *length*)

gzinflate décompresse une chaîne.

8.147.18 gzopen() : Ouvre un fichier compressé avec gzip

resource **gzopen** (string *filename* , string *mode* , int *use_include_path*)

gzopen ouvre un fichier compressé avec gzip pour y écrire ou y lire des données.

gzopen peut être utilisée pour lire un fichier qui n'est pas dans un format gzip ; dans ce cas, gzread liera directement le fichier sans décompression.

8.147.19 gzpassthru() : Affiche toutes les données qui restent dans un pointeur gz

int **gzpassthru** (resource *zp*)

gzpassthru lit les données restantes du fichier *zp* jusqu'à la fin, puis affiche le résultat (décompressé).

Note

Vous pourriez avoir besoin d'appeler la fonction <u>gzrewind</u> pour réinitialiser le pointeur de fichier au début du fichier si vous avez déjà écrit des données dedans.
--

Array

8.147.20 gzputs() : Alias de gzwrite

Cette fonction est un alias de : gzwrite .

8.147.21 gzread() : Lecture de fichier compressé binaire

string **gzread** (resource zp , int length)

gzread lit jusqu'à length octets dans le fichier compressé gzip, représenté par zp . La lecture s'arrête lorsque length octets (décompressés) ont été lus, ou que la fin du fichier a été atteinte.

8.147.22 gzrewind() : Replace le pointeur au début du fichier

bool **gzrewind** (resource zp)

gzrewind replace le pointeur de position du fichier zp au début de celui-ci.

8.147.23 gzseek() : Déplace le pointeur de lecture

int **gzseek** (resource zp , int offset)

gzseek place le pointeur de lecture du fichier zp à la position offset , comptée en octets depuis le début du fichier. C'est l'équivalent de la fonction gzseek(zp, offset, SEEK_SET) , du langage C.

Si le fichier est ouvert en lecture, cette fonction est alors émulée, et se révèle extrêmement lente. Si le fichier est ouvert en écriture, seuls les déplacements vers l'avant sont supportés : gzseek compresse alors une série de zéros jusqu'à la nouvelle position.

8.147.24 gztell() : Lit la position courante du pointeur de lecture

int **gztell** (resource zp)

gztell retourne la position du pointeur de lecture dans le fichier zp .

8.147.25 gzuncompress() : Décompresse une chaîne compressée

string **gzuncompress** (string data , int length)

gzuncompress décompresse une chaîne compressée.

8.147.26 gzwrite() : Ecrit dans un fichier compressé gzip

int **gzwrite** (resource zp , string string , int length)

gzwrite écrit le contenu de la chaîne string dans le fichier compressé zp .

8.147.27 readgzfile() : Lit tout le fichier compressé

int **readgzfile** (string filename , int use_include_path)

readgzfile lit le fichier filename , le décompresse et affiche le résultat.

readgzfile peut aussi servir à lire un fichier qui n'est pas compressé : dans ce cas, readgzfile va lire le fichier sans le décompresser.

8.147.28 zlib_get_coding_type() : Retourne la méthode de compression utilisée avec Gzip

string **zlib_get_coding_type** ()

zlib_get_coding_type retourne la méthode de compression utilisée avec Gzip.

9 PHP et le moteur interne Zend

Si vous êtes sur le point de développer des extensions PHP ou Zend, vous devez vous habituer à l'environnement de programmation fournissant diverses APIs. Cette partie de la documentation tente d'introduire les APIs fournies par les diverses versions du moteur PHP et Zend. Tant que les informations disponibles ici ne seront pas plus à jour, vous devriez lire les différents fichiers trouvés dans le code source de PHP, comme README.SELF-CONTAINED-EXTENSIONS et README.EXT_SKEL en complément du manuel.

9.1 API de flux : les auteurs d'extensions

9.1.1 Présentation

Les flux de PHP proposent une approche unifiée à la gestion des fichiers et sockets, depuis une extension PHP. En utilisant l'API standard, les flux vous permettent d'accéder à des fichiers, des sockets, des URL, de la mémoire ou encore des objets définis dans des scripts. Les flux (streams en anglais), sont une API flexible et dynamique, dynamiquement chargée dans les modules (et les scripts!).

Le but des flux est de proposer une API confortables pour les programmeurs, lors des manipulations de fichiers, URL et autres sources de données linéaires. L'API unifiée est très simple à comprendre : elle est plus ou moins basée sur la famille de fonction classiques C d'entrée/sortie (avec les mêmes concepts pour la plupart des fonctions) : les programmeurs C ne seront pas dépaysés.

L'API des flux intervient à différents niveaux : à la base, elle définit un objet `php_stream`, qui représente les sources de données. A un niveau au-dessus, l'API définit un objet `php_stream_wrapper` qui s'interface avec l'API de bas niveau, pour lire et écrire les données et méta données, dans les sources. Un paramètre additionnel `context`, accepté par la plupart des fonctions de flux, est passé à la méthode du gestionnaire `stream_opener` pour affiner le comportement du gestionnaire.

Tous les flux, une fois ouvert, peuvent également avoir n'importe quel nombre de filtres, traités sur les données lors de la lecture ou de l'écriture sur le flux.

Les flux peuvent être converti en pilotes pour fichiers, de manière à être utilisé par des bibliothèques tierce-partie avec beaucoup de facilité. Cela autorise ces bibliothèques à accéder aux données directement depuis des sources représentées par des URL. Si votre système supporte la fonction `opencookie`, vous pouvez même passer n'importe quel flux PHP à une autre bibliothèque qui utilise les entrée/sortie standard ANSI!

Note

Les fonctions de ce chapitre sont destinées à être utilisée dans le code source PHP et non pas dans les fonctions PHP. Les fonctions de flux, accessibles depuis un script, sont dans la section Flux .

9.1.2 Eléments de bases

Utiliser les flux ressemble énormément à l'utilisation des fonctions d'entrée/sortie standard ANSI streams. La principale différence est que vous devez créer le pointeur de flux d'abord. Dans la

plupart des cas, vous utiliserez la fonction `php_stream_open_wrapper` pour l'obtenir. Cette fonction ressemble à `fopen()`, comme le montre l'exemple ci-dessous :

```

Exemple de flux qui affiche la page de www.php.net
php_stream * stream = php_stream_open_wrapper("http://www.php.net", "rb", REPORT_ERRORS, NULL);
if (stream) {
    while(!php_stream_eof(stream)) {
        char buf[1024];

        if (php_stream_gets(stream, buf, sizeof(buf))) {
            printf(buf);
        } else {
            break;
        }
    }
    php_stream_close(stream);
}
    
```

La table ci-dessous montre les fonctions de flux équivalentes aux fonctions ANSI. Sauf indication contraire, les fonctionnalités sont les mêmes.

Fonction ANSI Stdio	Fonction Flux PHP	Notes
<code>fopen</code>	<code>php_stream_open_wrapper</code>	Les flux incluent des paramètres supplémentaires
<code>fclose</code>	<code>php_stream_close</code>	
<code>fgets</code>	<code>php_stream_gets</code>	
<code>fread</code>	<code>php_stream_read</code>	Le paramètre <code>nmemb</code> est supposé avoir la valeur 1 : le prototype ressemble plutôt à <code>read(2)</code>
<code>fwrite</code>	<code>php_stream_write</code>	Le paramètre <code>nmemb</code> est supposé avoir la valeur 1 : le prototype ressemble plutôt à <code>write(2)</code>
<code>fseek</code>	<code>php_stream_seek</code>	
<code>ftell</code>	<code>php_stream_tell</code>	
<code>rewind</code>	<code>php_stream_rewind</code>	
<code>feof</code>	<code>php_stream_eof</code>	
<code>fgetc</code>	<code>php_stream_getc</code>	
<code>fputc</code>	<code>php_stream_putc</code>	
<code>fflush</code>	<code>php_stream_flush</code>	
<code>puts</code>	<code>php_stream_puts</code>	Identique à <code>puts</code> , et non pas à <code>fputs</code>
<code>fstat</code>	<code>php_stream_stat</code>	Les informations des flux sont plus riches

9.1.3 Les flux qui sont des ressources

Tous les flux sont enregistrés comme des ressources lorsqu'ils sont créés. Cela garantit qu'ils seront convenablement déblayés de la mémoire, même en cas d'erreur fatale. Toutes les fonctions du système de fichier de PHP opèrent sur des ressources : cela signifie que vos extensions peuvent accepter des pointeurs de fichiers comme paramètre, et retourner des flux. L'API des flux rend ce processus très convivial.

```

Comment accepter un flux comme paramètre d'entrée
PHP_FUNCTION(example_write_hello)
{
    zval *zstream;
    php_stream *stream;

    if (FAILURE == zend_parse_parameters(ZEND_NUM_ARGS() TSRMLS_CC, "r", &zstream))
    
```

```

    return;

    php_stream_from_zval(stream, &zstream);

    /* Vous pouvez maintenant utiliser votre flux. Cependant, vous ne
       "possédez" pas le flux, mais le script le possède. Cela signifie
       que vous DEVEZ PAS fermer le flux, car sinon, PHP va crasher! */

    php_stream_write(stream, "hello\n");

    RETURN_TRUE();
}

```

Comment retourner un flux depuis une fonction

```

PHP_FUNCTION(example_open_php_home_page)
{
    php_stream *stream;

    stream = php_stream_open_wrapper("http://www.php.net", "rb", REPORT_ERRORS, NULL);

    php_stream_to_zval(stream, return_value);

    /* A partir de ce moment, le flux appartient au script.
       Si vous le fermez ici, vous allez faire crasher PHP! */
}

```

Comme les flux sont automatiquement déblayés de la mémoire, il est tentant de penser que vous pouvez être des programmeurs paresseux et ne pas vous préoccuper de la fermeture du flux lorsque vous en avez terminé avec lui. Même si une telle approche fonctionnera probablement, ce n'est pas une bonne idée pour un bon nombre de raisons : les flux conservent des verrous sur les ressources systèmes durant un temps plus long, ce qui fait que laisser le verrous empêche d'autres processus d'accéder à cette ressource. Si le script gère un grand nombre de fichiers, l'accumulation de ressources, en terme de mémoire et de nombre de fichiers ouverts, causer l'échec des requêtes du serveur web. Cela n'est plus aussi séduisant, n'est-ce pas? L'API de flux inclut un peu de magie pour que vous puissiez garder votre code propre : si un flux n'est pas fermé par votre code alors qu'il devrait l'être, vous trouverez des informations de débogage dans le fichier d'historique d'erreur de votre serveur web.

Note

Utilisez toujours une version de débogage de PHP lorsque vous développez (--enable-debug lors de la configuration de la compilation) car de nombreux efforts ont été fait pour vous aider à traquer les fuites de mémoire.

Dans certains cas, il est pratique de garder un flux ouvert pour toute la durée de la requête, afin de l'utiliser comme fichier d'historique. Ecrire le code nécessaire au nettoyage propre d'un tel fichier n'est pas difficile, mais ce sont plusieurs lignes de code qui ne sont pas strictement nécessaires. Pour vous éviter l'écriture de ce code, vous pouvez marquer le flux pour qu'il soit automatiquement libéré lorsque viendra le temps de libérer les ressources. Pour cela, vous pouvez utiliser la fonction `php_stream_auto_cleanup`.

9.2 internals.pdo

PDO Driver How-To

9.3 API Zend : Modification du coeur de PHP

Fonctions de l'API (travail en cours)

9.4 Étendre PHP 3

Cette section est très ancienne, et aborde surtout l'extension de PHP 3. Si vous êtes intéressés à PHP 4, lisez la section sur l' [API Zend](#) .

9.4.1 Créer une fonction PHP

9.4.1.1 Prototypes de fonctions

Toutes les fonctions suivent le schéma suivant :

```
void php3_foo(INTERNAL_FUNCTION_PARAMETERS) {
}
```

Même si votre fonction ne prend aucun argument, c'est comme cela qu'elle doit être appelée.

9.4.1.2 Arguments de fonctions

Les arguments sont toujours de type val. Ce type contient un membre de type union, qui indique le type réel de l'argument. De cette façon, si votre fonction prend deux arguments, elle ressemble à ceci :

Argument de fonction de lecture
<pre>pval *arg1, *arg2; if (ARG_COUNT(ht) != 2 getParameters(ht,2,&arg1,&arg2)==FAILURE) { WRONG_PARAM_COUNT; }</pre>

NOTE : Les arguments peuvent être passés par valeur ou par référence. Dans les deux cas, vous devez passer `&(pval *)` à `getParameters`. Si vous voulez vérifier que le n-ième paramètre a été passé par référence ou par valeur, vous devez utiliser la fonction `ParameterPassedByReference(ht,n)`. Elle retournera 1 ou 0.

Lorsque vous modifiez l'un des paramètres, qu'ils soient envoyés par référence ou par valeur, vous pouvez le passer à `pval_destructor` pour le réinitialiser, ou, s'il s'agit d'un tableau et que vous voulez ajouter des valeurs, vous pouvez utiliser des fonctions similaires à celles qui sont dans `internal_functions.h`, qui manipule `return_value` comme tableau.

Par ailleurs, si vous modifiez un paramètre en `IS_STRING`, assurez-vous que vous avez bien assigné une nouvelle chaîne avec `estrdup()` et une nouvelle longueur de chaîne. Seulement après, vous pouvez modifier le type en `IS_STRING`. Si vous modifiez une chaîne en `IS_STRING` ou `IS_ARRAY` vous devez d'abord appeler le destructeur `pval_destructor`.

9.4.1.3 Fonctions à nombre d'arguments variable

Une fonction peut prendre un nombre variable d'arguments. Si votre fonction peut prendre deux ou trois arguments, utiliser la syntaxe suivante :

Fonctions à nombre d'arguments variable
<pre>pval *arg1, *arg2, *arg3; int arg_count = ARG_COUNT(ht); if (arg_count < 2 arg_count > 3 getParameters(ht, arg_count, &arg1, &arg2, &arg3) == FAILURE) { WRONG_PARAM_COUNT; }</pre>

9.4.1.4 Utiliser les arguments d'une fonction

Le type de chaque argument est stocké dans le champ pval. Ce champ peut prendre les valeurs suivantes :

IS_STRING	Chaîne de caractères
IS_DOUBLE	Nombre à virgule flottante, en précision double
IS_LONG	Entier long
IS_ARRAY	Tableau
IS_EMPTY	Aucune
IS_USER_FUNCTION	??
IS_INTERNAL_FUNCTION	?? (Si ce type ne peut pas être passé à une fonction, effacez-le)
IS_CLASS	??
IS_OBJECT	??

Si vous recevez un argument d'un type, et que vous voulez l'utiliser avec un autre type, ou si vous voulez simplement forcer le type, vous pouvez utiliser l'une des fonctions de conversion suivantes :

```
convert_to_long(arg1);
convert_to_double(arg1);
convert_to_string(arg1);
convert_to_boolean_long(arg1); /* Si la chaîne est "" ou "0" elle devient 0, 1 sinon */
convert_string_to_number(arg1); /* Convertit une chaîne en LONG ou DOUBLE suivant la chaîne */
```

Ces fonctions convertissent sur place : elles ne retournent aucune valeur.

La valeur de l'argument est enregistrée dans une union. Les membres sont :

- IS_STRING: arg1->value.str.val
- IS_LONG: arg1->value.lval
- IS_DOUBLE: arg1->value.dval

9.4.1.5 Gestion de la mémoire dans une fonction

Toute la mémoire nécessaire à une fonction doit être allouée avec `emalloc()` ou `estrdup()`. Ces fonctions ont le goût et l'odeur des fonctions C classiques `malloc()` et `strdup()`. La mémoire doit être libérée avec `efree()`.

Il y a deux types de mémoire dans ce programme : la mémoire qui est retournée à l'analyseur, et la mémoire qui est nécessaire pour le stockage temporaire dans la fonction. Lorsque vous assignez une

chaîne dans une variable qui est retournée à l'analyseur, assurez-vous de bien allouer la mémoire avec `emalloc()` ou `estrdup()`. Cette mémoire ne doit JAMAIS être libérée, sauf si vous réécrivez votre original plus loin, dans la même fonction (mais ce n'est pas de la programmation propre).

Pour tous vos besoins en mémoire temporaire/permanente dont vous avez besoin dans vos fonctions/bibliothèques, vous devez utiliser les fonctions `emalloc()`, `estrdup()` et `efree()`. Elles se comportent EXACTEMENT comme leurs homologues. Tout ce qui est créé avec `emalloc()` ou `estrdup()` doit être libéré avec `efree()` à un moment ou un autre, à moins que ce ne soit utile ailleurs dans le programme; sinon, il va y avoir une fuite de mémoire. La signification de "Elles se comportent EXACTEMENT comme leurs homologues" est que si vous libérez une variable qui n'a pas été créée avec `emalloc()` ou `estrdup()`, vous courez droit à au crash ("segmentation fault"). Soyez alors extrêmement prudent, et libérez toute votre mémoire inutilisée.

Si vous compilez avec "-DDEBUG", PHP 3 affichera la liste de tous les appels à `emalloc()` et `estrdup()` mais jamais à `efree()` lorsque celui-ci intervient dans un script spécifié.

9.4.1.6 Affecter une variable dans la table des symboles

Un grand nombre de macros sont disponibles pour rendre plus facile l'insertion de variables dans la table des symboles :

- `SET_VAR_STRING(name,value)`
- `SET_VAR_DOUBLE(name,value)`
- `SET_VAR_LONG(name,value)`

Attention

Soyez prudent avec la constante `SET_VAR_STRING`. La valeur doit être placée dans une portion de mémoire créée avec `malloc()` , sinon le gestionnaire de mémoire essaiera de libérer le pointeur plus tard. Ne passez aucune mémoire allouée statiquement à `SET_VAR_STRING`.

Les tables des symboles de PHP est une table de hash. A n'importe quel moment, `&symbol_table` est un pointeur sur la table principale, et `active_symbol_table` pointe sur la table actuellement utilisée. (ces deux tables peuvent être identiques au démarrage, ou différent, suivant que vous êtes dans une fonction ou non).

Les exemples suivants utilisent 'active_symbol_table'. Vous devriez la remplacer par `&symbol_table` si vous voulez travailler sur la table principale. De plus, les mêmes fonctions peuvent être appliquées à des tableaux, comme expliqué ci-dessous.

Vérification de l'existence de \$foo dans la table des symboles

```
if (hash_exists(active_symbol_table, "foo", sizeof("foo"))) { existe... }
else { n'existe pas }
```

Rechercher la taille d'une variable dans la table des symboles

```
hash_find(active_symbol_table, "foo", sizeof("foo"), &pvalue);
check(pvalue.type);
```

En PHP, les tableaux sont implémentés en utilisant les mêmes tables de hash que les variables. Cela signifie que les deux fonctions ci-dessus peuvent être appelées pour vérifier la présence de variables dans un tableau.

Si vous voulez définir un nouveau tableau dans la table des symboles, utilisez le code suivant.

D'abord, vous devez vérifier qu'il n'existe pas, avec `hash_exists()` ou `hash_find()`.

Puis, initialisez le tableau :

Initialisation d'un tableau

```
pval arr;

if (array_init(&arr) == FAILURE) { Echec de l'initialiation... };
hash_update(active_symbol_table, "foo", sizeof("foo"), &arr, sizeof(pval), NULL);
```

Ce code déclare un nouveau tableau, appelé \$foo , dans la table de symbole. Ce tableau est vide.

Voici comment ajouter deux nouvelles entrées dans ce tableau :

Ajout d'entrées dans un tableau

```
pval entry;

entry.type = IS_LONG;
entry.value.lval = 5;

/* définit $foo["bar"] = 5 */
hash_update(arr.value.ht, "bar", sizeof("bar"), &entry, sizeof(pval), NULL);

/* définit $foo[7] = 5 */
hash_index_update(arr.value.ht, 7, &entry, sizeof(pval), NULL);

/* définit la prochaine place libre dans $foo[],
 * $foo[8], qui sera 5 (comme en php2)
 */
hash_next_index_insert(arr.value.ht, &entry, sizeof(pval), NULL);
```

Si vous voulez modifier une valeur que vous avez insérée dans une table de hash, vous devez d'abord la lire dans la table. Pour éviter cette recherche, vous pouvez fournir une pval ** à la fonction d'ajout dans la table de hash, et elle modifiera la valeur à l'adresse pval *, avec la valeur donnée. Si cette valeur est NULL , (comme dans tous les exemples ci-dessus), ce paramètre sera ignoré.

hash_next_index_insert() utiliser plus ou moins la même logique que \$foo[] = bar; in PHP 2.0.

Si vous construisez un tableau, pour le retourner, vous pouvez l'initialiser comme ceci :

```
if (array_init(return_value) == FAILURE) { échec...; }
```

...puis ajouter les valeurs grâce aux macros :

```
add_next_index_long(return_value, long_value);
add_next_index_double(return_value, double_value);
add_next_index_string(return_value, estrdup(string_value));
```

Bien sûr, si l'ajout n'est pas fait juste après l'initialisation, vous devrez d'abord rechercher le tableau :

```
pval *arr;
```

```
if (hash_find(active_symbol_table, "foo", sizeof("foo"), (void **) &arr) == FAILURE) { introuvable... }
else { utilisez arr->value.ht... }
```

Notez que hash_find reçoit un pointeur sur un pointeur sur pval, et pas un pointeur sur pval.

Toutes les fonctions d'accès aux hash retourne TRUE (SUCCES) ou FALSE (FAILURE), excepté hash_exists(), qui retourne un booléen.

9.4.1.7 Retourne une valeur simple

Un grand nombre de macros sont disponible pour simplifier le retour des valeurs.

La macro RETURN_* fixe la valeur de retour, et termine la fonction :

- RETURN
- RETURN_FALSE
- RETURN_TRUE
- RETURN_LONG(l)
- RETURN_STRING(s,dup) Si dup est TRUE , duplique la chaîne.
- RETURN_STRINGL(s,l,dup) Retourne la chaîne (s) en spécifiant la longueur (l).
- RETURN_DOUBLE(d)

La macro RETVAL_* macros fixe la valeur de retour, mais ne termine pas la fonction :

- RETVAL_FALSE
- RETVAL_TRUE
- RETVAL_LONG(l)
- RETVAL_STRING(s,dup) Si dup est TRUE , duplique la chaîne.
- RETVAL_STRINGL(s,l,dup) Retourne la chaîne (s) en spécifiant la longueur (l).
- RETVAL_DOUBLE(d)

Les macros ci-dessus vont utiliser estrdup() sur les arguments passés. Cela vous permet de libérer tranquillement les arguments après avoir appelé cette fonction, ou bien, utiliser de la mémoire allouée statiquement.

Si votre fonction retourne un booléen de succès/erreur, utilisez toujours RETURN_TRUE et RETURN_FALSE respectivement.

9.4.1.8 Retourner des valeurs complexes

Votre fonction peut aussi retourner des valeurs complexes, tels que des objets ou tableaux.

Retourner un objet :

1. Appeler object_init(return_value).
2. Remplissez les valeurs. Les fonctions utilisables sont listées ci dessous.
3. Éventuellement, enregistrez les fonctions pour cet objet. Afin de lire des valeurs de cet objet, la fonction doit lire dans "this", dans la table de symbole active active_symbol_table. Son type doit être IS_OBJECT, et c'est une table de hash basique. (i.e., vous pouvez utiliser les fonctions habituelles de .value.ht). L'enregistrement réel peut être fait comme suit :

```
add_method( return_value, function_name, function_ptr );
```

Les fonctions d'accès aux objets sont :

- add_property_long(return_value, property_name, l) - Ajoute un membre nommé 'property_name', de type long, égal à 'l'
- add_property_double(return_value, property_name, d) - Idem, ajoute un double
- add_property_string(return_value, property_name, str) - Idem, ajoute une chaîne
- add_property_stringl(return_value, property_name, str, l) - Idem, ajoute une chaîne de longueur 'l'

Retournez un tableau :

1. Appelez `array_init(return_value)`.
2. Remplissez les valeurs. Les fonctions disponibles sont listées ci-dessous.

Les fonctions utilisées pour accéder à un tableau sont :

- `add_assoc_long(return_value,key,l)` - Ajoute une entrée associative avec la clé 'key' et la valeur 'l', de type long
- `add_assoc_double(return_value,key,d)`
- `add_assoc_string(return_value,key,str,duplicate)`
- `add_assoc_stringl(return_value,key,str,length,duplicate)` spécifie la taille d'une chaîne
- `add_index_long(return_value,index,l)` - Ajoute une entrée d'index 'index' avec la valeur 'l', de type long
- `add_index_double(return_value,index,d)`
- `add_index_string(return_value,index,str)`
- `add_index_stringl(return_value,index,str,length)` - spécifie la longueur de la chaîne.
- `add_next_index_long(return_value,l)` - ajoute une entrée tableau, dans le prochain offset libre, de longueur 'l', de type long
- `add_next_index_double(return_value,d)`
- `add_next_index_string(return_value,str)`
- `add_next_index_stringl(return_value,str,length)` - spécifie la taille d'une chaîne

9.4.1.9 Utilisation de la liste des ressources

PHP dispose de standards pour traiter un certains nombre de ressources. Ils remplacent tous les listes de PHP 2.0.

Fonctions accessibles :

- `php3_list_insert(ptr, type)` - retourne l'identifiant 'id' de la nouvelle ressource insérée.
- `php3_list_delete(id)` - efface la ressource d'identifiant id
- `php3_list_find(id,*type)` - retourne le pointeur de la ressource d'identifiant id, et modifie le type 'type'

Typiquement, ces fonctions sont utilisées pour les pilotes SQL, mais elles peuvent servir n'importe quoi d'autre. Par exemple, conserver un pointeur de fichier.

La liste standard de code ressemble à ceci :

Ajouter une nouvelle ressource
<pre>RESOURCE *resource; /* ...alloue de la mémoire pour la ressource, et l'acquiert ... */ /* Ajoute la nouvelle ressource dans la liste */ return_value->value.lval = php3_list_insert((void *) resource, LE_RESOURCE_TYPE); return_value->type = IS_LONG;</pre>
Utiliser une ressource existante
<pre>pval *resource_id; RESOURCE *resource; int type; convert_to_long(resource_id); resource = php3_list_find(resource_id->value.lval, &type); if (type != LE_RESOURCE_TYPE) {</pre>

```

    php3_error(E_WARNING, "la ressource dont l'index est %d a un type incorrect", resource_id->value);
    RETURN_FALSE;
}
/* ...utilisation de la ressource... */

```

Effacer une ressource existante

```

pval *resource_id;
RESOURCE *resource;
int type;

convert_to_long(resource_id);
php3_list_delete(resource_id->value.lval);

```

Les types de ressources doivent être enregistré dans le fichier `php3_list.h`, dans l'énumération `list_entry_type`. En plus, il faut penser à ajouter une fonction de terminaison, pour chaque type de ressource défini, dans le fichier `list.c`, pour la fonction `list_entry_destructor()` (même si vous n'avez rien de particulier à faire lors de la terminaison, vous devez au moins ajouter un cas vide).

9.4.1.10 Utilisation de la table des ressources persistantes

PHP dispose d'un lieu de stockage des ressources persistantes (i.e., les ressources qui doivent être conservées d'un hit à l'autre). Le premier module à utiliser cette capacité a été MySQL, et mSQL suivi, ce qui fait que l'on peut se faire une impression du fonctionnement de cette fonction avec `mysql.c`. Les fonctions ressemblent à ceci :

- `php3_mysql_do_connect`
- `php3_mysql_connect()`
- `php3_mysql_pconnect()`

L'idée conductrice de ces modules est la suivante :

1. Programmez tout votre module pour qu'il travaille avec les ressources standard, comme mentionné dans la section (9).
2. Ajoutez une autre fonction de connexion, qui vérifie d'abord que la ressource existe dans la liste des ressources persistantes. Si c'est le cas, enregistrez cette ressource comme pour les ressources standard (et grâce à la première étape, cela va fonctionner immédiatement). Si la ressource n'existe pas, créez la, ajoutez la à la liste de ressources persistantes, et ajoutez la à la liste de ressources, ce qui fait que le code va fonctionner, et que le prochain appel renverra une ressource existante. Vous devez enregistrer ces fonctions avec un type différent (`LE_MYSQL_LINK` pour les liens non persistants, et `LE_MYSQL_PLINK` pour les liens persistants).

Si vous jetez un oeil dans `mysql.c`, vous verrez que, hormis la fonction de connexion complexe, rien n'a du être changé dans le module.

La même interface existe pour la liste des ressources standard, et pour la liste des ressources persistantes, seule la 'list' est remplacée par 'plist':

- `php3_plist_insert(ptr, type)` - retourne l'identifiant 'id' de la nouvelle ressource insérée
- `php3_plist_delete(id)` - efface la ressource d'identifiant id
- `php3_plist_find(id,*type)` - retourne le pointeur de la ressource d'identifiant id, et modifie le type 'type'

Cependant, il est probable que ces fonctions seront inutiles pour vous, lorsque vous essayerez d'implémenter un module persistant. Typiquement, on utilise le fait que la liste de ressources persistantes est une table de hash. Par exemple, dans les modules MySQL/mSQL, lors d'un appel à `pconnect()`, la fonction construit une chaîne avec l'hôte/utilisateur/mot_de_passe, et l'utilise pour

enregistrer dans la table de hash. Au prochain appel, avec les mêmes hôte/utilisateur/mot_de_passe, la même clé sera générée, et la ressource associée sera retrouvée.

Jusqu'à ce que la documentation s'étoffe, jetez un oeil aux fichiers mysql.c ou msql.c pour voir comment implémentez vos accès aux ressources persistantes.

Une chose importante à noter : les ressources qui sont enregistrées dans la liste de ressource persistante ne DOIVENT PAS être allouée avec le gestionnaire de mémoire PHP, c'est-à-dire qu'elles ne doivent pas être créée avec `emalloc()`, `estrdup()`, etc. Au contraire, il faut utiliser les fonctions standard `malloc()`, `strdup()`, etc. La raison est for simple : à la fin de la requête, la mémoire sera supprimée par le gestionnaire. Étant donné que les liens persistants doivent être conservés, il ne faut pas utiliser le gestionnaire de mémoire.

Lorsque vous enregistrez une ressource qui sera placé dans la liste de ressources persistantes, il faut ajouter les destructeurs dans les deux listes de ressources, persistantes ou pas. Le destructeur de la liste de ressources non persistantes ne doit rien faire du tout, tandis que celui de la liste de ressources persistantes doit libérer proprement toutes les ressources acquises (mémoire, lien SQL, etc.). Comme pour les ressources non persistantes vous DEVEZ ajouter un destructeur, même s'il ne fait rien. N'oubliez pas que `emalloc()` et compagnie ne doivent pas être utilisé en conjonction avec la liste de ressources persistantes, et donc, vous ne devez pas utiliser `efree()` non plus.

9.4.1.11 Ajouter des directives de configuration à l'exécution

De nombreuses caractéristiques de PHP peuvent être configurée à l'exécution. Ces directives peuvent apparaître dans le fichier `php3.ini`, ou, dans le cas du module Apache, dans le fichier `.conf`. L'avantage de l'avoir dans le fichier `.conf`, est que ces caractéristiques peuvent être configurées dossier par dossier. Cela signifie qu'un dossier peut avoir un safe mode `exec` dir, tandis qu'un autre en aura un autre. Cette granularité de la configuration peut être extrêmement pratique lorsque le serveur supporte plusieurs serveurs virtuels.

Les étapes de configuration d'une nouvelle directive sont :

1. Ajouter la directive à la structure `php3_ini_structure` dans le fichier `mod_php3.h`.
2. Dans `main.c`, éditez la fonction `php3_module_startup` et ajoutez l'appel approprié à `cfg_get_string()` ou `cfg_get_long()`.
3. Ajoutez la directive, ses restrictions et un commentaire dans la structure `php3_commands` du fichier `mod_php3.c`. Notez la partie restrictions `RSRC_CONF` sont des directives qui ne peuvent être disponibles que dans le fichier de configuration Apache. Toutes les directives `OR_OPTIONS` peuvent être placées n'importe où, y compris dans un fichier `.htaccess`.
4. Soit dans `php3take1handler()`, soit dans `php3flaghandler()`, ajoutez l'entrée appropriée pour votre directive.
5. Dans la section de configuration, de `_php3_info()`, dans le fichier `functions/info.c`, vous devez ajouter votre configuration.
6. Finalement, vous devez utiliser votre configuration quelque part. Elle sera accessible par `php3_ini.directive`.

9.4.2 Appeler des fonctions utilisateurs

Pour appeler des fonctions utilisateurs depuis une fonction interne, vous devez utiliser la fonction `call_user_function`.

`call_user_function` retourne SUCCESS en cas de succès, et FAILURE en cas d'échec, ou si la fonction n'a pas été trouvée. Vous devez vérifier cette valeur. Si la réponse est SUCCESS, vous êtes responsable de la destruction de `retval` (ou alors, retournez la comme valeur de réponse de votre fonction). Si la réponse est FAILURE, la valeur de `retval` est indéfinie, et vous ne devez pas y toucher.

Toutes les fonctions internes qui appellent une fonction utilisateur, **DOIVENT** être réentrante. En particulier, elles ne doivent pas utiliser de valeurs globales, ou de variables statiques.

`call_user_function` prend 6 arguments :

9.4.2.1 HashTable *function_table

La table de hash dans laquelle la fonction doit être recherchée.

9.4.2.2 pval *object

Un pointeur sur un objet sur lequel la fonction est invoquée. Il devrait être à NULL, si on invoque une fonction globale. S'il n'est pas à NULL (c'est-à-dire, il pointe sur un objet), l'argument `function_table` est ignoré, et la liste des fonctions sera lue dans l'objet, plutôt que dans l'argument. L'objet PEUT être modifié par la fonction qui est appelée (la fonction y aura accès via `$this`). Si, pour quelque raison, vous ne le voulez pas, envoyez une copie de l'objet à la place.

9.4.2.3 pval *function_name

Le nom de la fonction à appeler. Elle doit être de type pval, IS_STRING, avec les valeurs de `function_name.str.val` et `function_name.str.len` correctes. `function_name` est modifié par `call_user_function()` - il est converti en minuscule. Si vous voulez préserver la casse, envoyez une copie du nom de la fonction.

9.4.2.4 pval *retval

Un pointeur sur une structure pval, dans laquelle la valeur de retour de la fonction sera placée. La structure doit avoir été allouée au préalable - `call_user_function` ne l'allouera pas.

9.4.2.5 int param_count

Le nombre de paramètre passé à la fonction.

9.4.2.6 pval *params[]

Un tableau de pointeur sur les valeurs qui vont être passées comme arguments à la fonction. Le premier argument est à l'offset 0, le second à l'offset 1, ... Le tableau est un tableau de pointeurs sur pval; Les pointeurs sont envoyés tels quels à la fonction, ce qui signifie que si la fonction modifie les arguments, les valeurs originales seront modifiées. Si vous voulez l'éviter, passez une copie à la place.

9.4.3 Rapport d'erreurs

Pour signaler les erreurs d'une fonction interne, vous devez appeler la fonction `php3_error`. Cette fonction prend deux arguments au moins : le niveau de l'erreur, et le message d'erreur, sous forme de chaîne de caractères formatée (en tant qu'un appel standard `printf`). Tous les arguments suivants sont des paramètres de formats de chaîne. Les niveaux d'erreurs sont :

9.4.3.1 E_NOTICE

Les notes ne sont pas affichées par défaut, et indiquent que le script a rencontré quelque chose qui peut être une erreur, mais peut aussi être un événement normal dans la vie du script. Par exemple, essayer d'accéder à une valeur qui n'a pas été déclarée, ou appeler `stat` sur un fichier qui n'existe pas.

9.4.3.2 E_WARNING

Les alertes sont affichées par défaut, mais n'interrompent pas l'exécution du script. Elles indiquent un problème qui doit être intercepté par le script avant que l'appel. Par exemple, appeler `ereg` avec une regex invalide.

9.4.3.3 E_ERROR

Les erreurs sont aussi affichées par défaut, et l'exécution du script est interrompue. Elles indiquent des erreurs qui ne peuvent pas être ignorées, comme des problèmes d'allocation de mémoire, par exemple.

9.4.3.4 E_PARSE

Les erreurs d'analyse doivent être générées que par l'analyseur. Elles ne sont citées ici que dans le but d'être exhaustif.

9.4.3.5 E_CORE_ERROR

Elles sont similaires aux erreurs `E_ERROR`, mais elles sont générées par le code de PHP. Les fonctions ne doivent pas générer ce genre d'erreur.

9.4.3.6 E_CORE_WARNING

Elles sont similaires à `E_WARNING`, mais elles sont générées par le code de PHP. Les fonctions ne doivent pas générer ce genre d'erreur.

9.4.3.7 E_COMPILE_ERROR

Elles sont similaires à `E_ERROR`, mais elles sont générées par Zend Scripting Engine. Les fonctions ne doivent pas générer ce genre d'erreur.

9.4.3.8 E_COMPILE_WARNING

Elles sont similaires à E_WARNING , mais elles sont générées par Zend Scripting Engine. Les fonctions ne doivent pas générer ce genre d'erreur.

9.4.3.9 E_USER_ERROR

E_USER_ERROR est comparable à E_ERROR . Elle est générée en PHP par l'utilisation de la fonction trigger_error . Les fonctions ne doivent pas générer ce genre d'erreur.

9.4.3.10 E_USER_WARNING

E_USER_WARNING est comparable à E_WARNING . Elle est générée en PHP par l'utilisation de la fonction trigger_error . Les fonctions ne doivent pas générer ce genre d'erreur.

9.4.3.11 E_USER_NOTICE

E_USER_WARNING est comparable à E_NOTICE . Elle est générée en PHP par l'utilisation de la fonction trigger_error . Les fonctions ne doivent pas générer ce genre d'erreur.

9.4.3.12 E_ALL

Toutes les erreurs ci-dessus. Utiliser ce niveau de rapport d'erreur provoque l'affichage de toutes les erreurs.

10 FAQ : foire Aux Questions

10.1 Informations générales

Cette section présente les questions les plus fréquentes à propos de PHP : ce qu'est PHP et ce qu'il fait.

10.1.1 Qu'est-ce que PHP ?

Depuis la [préface de ce manuel](#) :

PHP , est un acronyme récursif, qui signifie "PHP: Hypertext Preprocessor" : c'est un langage de script HTML, exécuté coté serveur. Sa syntaxe est empruntée aux langages C, Java et Perl, et est facile à apprendre. Le but de ce langage est de permettre aux développeurs web d'écrire des pages dynamiques rapidement, mais vous pouvez faire beaucoup plus avec PHP.

Une introduction à PHP par Stig Sæther Bakken est disponible à <http://www.zend.com/zend/art/intro.php> (en anglais) sur le site de Zend. En outre, une grande partie du [matériel de conférence de PHP](#) est librement disponible.

10.1.2 Que représente l'acronyme PHP ?

PHP signifie **PHP: Hypertext Preprocessor** . La confusion vient du fait que la première lettre de l'acronyme représente l'acronyme lui-même. Ce type d'acronyme est appelé un acronyme récursif. Les plus curieux peuvent visiter " [Free On-Line Dictionary of Computing](#) " pour plus d'informations sur les acronymes récursifs.

10.1.3 Quelle est la relation entre les versions ?

PHP/FI 2.0 est une des premières versions de PHP et elle n'est plus supportée. PHP 3 en est son successeur et est beaucoup plus conviviale. PHP 4 est la génération actuelle de PHP, qui utilise le [moteur Zend](#) pour le traitement. PHP 5 utilise le [moteur Zend 2](#) et apporte, entre autres, beaucoup de nouveautés dans le [model objet](#) .

10.1.4 Puis-je utiliser plusieurs versions de PHP en même temps ?

Oui. Voir le fichier INSTALL inclus dans les sources de PHP 4. Lisez également cet [annexe](#) .

10.1.5 Quelle est la différence entre PHP 3 et PHP 4 ?

Il y a beaucoup [d'articles](#) (anglais) relatifs à ce sujet écrits par les auteurs de PHP 4. Voici une liste des plus importantes nouvelles fonctionnalités :

- Module étendu de l'API
- Généralisation du processus d'installation sur les environnements Unix
- Une interface web générique qui supporte également les serveurs web multi-threadé

- Amélioration de la coloration syntaxique
- Support natif des sessions HTTP
- Support des tampons de sortie
- Système de configuration plus performant
- Le compte de référence

Merci de vous référer à l'article de Zend " [Qui a-t-il de nouveau en PHP 4 ?](#) " (anglais) pour plus de détails sur ces nouveautés et bien plus encore. Si vous migrez de PHP 3 à PHP 4, lisez également cet [annexe](#) .

10.1.6 Je pense avoir trouver un bogue ! Qui dois-je contacter ?

Vous devriez aller sur la base de données de bogues PHP afin de vous assurer qu'il n'est pas déjà connu. Si vous ne le trouvez pas, utilisez le formulaire de rapport de bogues pour le faire connaître. Il est important d'utiliser la base de données de bogues au lieu d'envoyer simplement un courriel à une des listes de diffusion car le bogue se verra assigner un numéro qui vous sera utile pour suivre son évolution. La base de données de bogues peut être trouvée à <http://bugs.php.net/> .

10.2 Listes de diffusions

Cette section traite des questions à propos de la communication avec le communauté PHP. Le meilleur moyen est l'utilisation des listes de diffusions.

10.2.1 Existe-t-il des listes de diffusions PHP ?

Bien sûr ! Il y a beaucoup de listes de diffusions traitant de baucoups de sujets. Une liste complète des listes de diffusions peut être trouvée sur notre [page de support](#) .

La liste de diffusions la plus générale est la liste php-general . Pour s'y inscrire, envoyez un courriel à php-general-subscribe@lists.php.net . Vous n'avez pas besoin d'inclure quoi que ce soit dans l'objet ou le corps du message. Pour se désinscrire, envoyez un courriel à php-general-unsubscribe@lists.php.net .

Vous pouvez également vous inscrire et vous désinscrire en utilisant l'interface web sur notre [page de support](#) .

10.2.2 Y a-t-il d'autre communauté ?

Il y en a énormément à travers le monde. Nous en avons recenser plusieurs, comme par exemple, quelques canaux IRC et des listes de diffusions en langues étrangères sur notre [page de support](#) .

10.2.3 Aidez-moi ! Je ne peux visiblement pas m'inscrire/désinscrire à une/d'une des listes de diffusions !

Si vous avez des problèmes pour l'inscription/la désinscription depuis la liste de diffusions php-general , cela peut être dû au fait que le logiciel de listes de diffusions n'arrive pas à traiter correctement la liste de diffusions à utiliser. Si vous adresse mèle est joeblow@example.com , vous pouvez envoyer une requête d'inscription à

php-general-subscribe-joeblow=example.com@lists.php.net , ou une requête de désinscription à php-general-unsubscribe-joeblow=example.com@lists.php.net . Utilisez la même méthode avec les autres listes de diffusions.

10.2.4 Y a-t-il une archive des listes de diffusions quelque part ?

Oui, vous pouvez la trouver sur le site d'archives des listes de diffusions sur la [page de support](#) . Les articles de la liste de diffusions sont aussi archivés comme des messages d'actualité. Vous pouvez y accéder à <news://news.php.net/> avec un client de news. Il y a également une interface web expérimentale pour les lire à <http://news.php.net/> .

10.2.5 Que puis-je demander sur les listes de diffusions ?

PHP devient de plus en plus populaire chaque jours, augmentant ainsi le trafic sur la liste de diffusions php-general qui accueillent environs 150 à 200 messages par jours. A cause de cela, vous ne devriez utiliser la liste de diffusions qu'en dernier ressort une fois avoir cherché partout ailleurs.

Avant de poster sur cette liste, merci de lire la FAQ (Foire Aux Questions) afin de voir si vous n'y trouvez pas de l'aide. S'il n'y a rien concernant votre problème, essayez dans les archives de la liste de diffusions (voir précédemment). Si vous avez des problèmes pour l'installation ou la configuration de PHP, lisez toute la documentation s'y rapportant, ainsi que les fichiers README . Si vous ne trouvez toujours pas de réponses à votre problème, vous êtes le bienvenu sur la liste de diffusions.

Avant de poser une question, vous devriez lire l'article intitulé "[Comment poser une question de la meilleure façon](#)" ; c'est une bonne idée, pour tout le monde.

10.2.6 Quelle information devrais-je inclure lors de l'écriture de mon message à la liste de diffusions ?

Les messages comme "Je n'arrive pas à faire fonctionner PHP !! Aidez moi !! Qu'est-ce qu'il ne va pas ??" sont totalement inutiles, pour tous. Si vous avez des problèmes pour faire fonctionner PHP, vous devez inclure le système sur lequel le problème survient, quelle version de PHP vous tenter de faire fonctionner, sous quelle forme vous l'avez installé (pré-compilé, CVS, RPMs, etc.), qu'est-ce que vous avez fait pour le moment, l'endroit où vous êtes coincé et quel est le message d'erreur.

Cela est valable pour n'importe quel type de problème. Vous devez inclure les informations sur ce que vous avez fait, où vous êtes coincé, ce que vous cherché à faire et, si possible, le message d'erreur obtenu. Si vous avez un problème avec votre code source, vous devez également inclure la portion de code qui ne fonctionne pas. N'incluez que le code nécessaire à la compréhension du problème ! Si vous incluez tout le code, votre message sera difficile à lire et beaucoup de personnes passeront sans même le lire à cause de cela. Si vous n'êtes pas sûr sur la quantité d'informations à inclure à votre message, il est préférable d'en inclure trop que pas assez.

Un autre point important à ne pas oublier est de résumer votre problème dans l'objet du message. Un objet qui ressemble à "Aidez moi !!!!" ou "Quel est le souci ici ?" sera ignoré par la majorité des lecteurs.

Et pour finir, vous êtes encouragé à lire l'article sur "[Comment poser une question de la meilleure façon](#)", ce qui sera une aide énorme pour tous, et tout spécialement pour vous.

10.3 Obtenir PHP

Cette section traite du téléchargement de PHP et les problèmes liés aux systèmes d'exploitation.

10.3.1 Où puis-je obtenir PHP ?

Vous pouvez télécharger PHP à partir d'un des membres du réseau de sites PHP. Vous pouvez les trouver sur <http://www.php.net/>. Vous pouvez aussi utiliser CVS pour obtenir la toute dernière version des sources. Pour plus d'informations, allez sur <http://www.php.net/anoncv.php>.

10.3.2 Est-ce que les versions binaires pré-compilées sont disponibles ?

Nous ne les distribuons que pour le système Windows, car nous ne pouvons compiler PHP pour chaque plate-forme Linux/Unix avec toutes les combinaisons d'extensions. Notez aussi que plusieurs distributions Linux fournissent PHP d'office de nos jours. Les binaires Windows peuvent être téléchargés à partir de notre page de [Téléchargement](#), pour les binaires Linux, visitez le site de votre distribution.

10.3.3 Où puis-je obtenir les bibliothèques dont j'ai besoin pour compiler les extensions optionnelles de PHP ?

Note
Celles marquées avec un * ne sont pas des bibliothèques thread-safe, et ne doivent pas être utilisées avec PHP en module de serveur dans les serveurs web Windows multi-threadés (IIS, Netscape). Cela n'est pas applicable pour les environnements Unix, pour le moment.

- [LDAP \(Unix\)](#) .
- [LDAP \(Unix/Win\)](#) : Mozilla Directory (LDAP) SDK
- [free LDAP server](#) .
- [Berkeley DB2 \(Unix/Win\)](#) : <http://www.sleepycat.com/>.
- [SNMP*](#) (Unix): .
- [GD*](#) (Unix/Win) .
- [mSQL*](#) (Unix) .
- [PostgreSQL \(Unix\)](#) .
- [IMAP*](#) (Win/Unix) .
- [Sybase-CT*](#) (Linux, libc5) : Disponible localement.
- [FreeType \(libtff\)](#) : .
- [ZLib \(Unix/Win32\)](#) .
- [expat XML parser \(Unix/Win32\)](#) .
- [PDFLib](#) .
- [mcrypt](#) .
- [mhash](#) .
- [t1lib](#) .
- [dmalloc](#) .
- [aspell](#) .

- [readline](#) .

10.3.4 Comment faire fonctionner ces bibliothèques ?

Vous devrez suivre les instructions fournies avec les bibliothèques. Quelques une d'entre-elles sont détectées automatiquement lorsque vous exécutez le script 'configure' de PHP (comme la bibliothèque GD), pour les autres, vous devrez les activer en utilisant l'option ' --with-EXTENSION ' de ' configure '. Exécutez ' configure --help ' pour en avoir la liste complète.

10.3.5 J'ai la dernière version du code source de PHP, téléchargé à partir de CVS. De quoi ai-je besoin pour le compiler sous Windows ?

Premièrement, vous avez besoin de Microsoft Visual C++ v6 (la v5 peut fonctionner, mais nous le faisons avec v6), et vous avez besoin de quelques fichiers de support. Voyez la section du manuel à propos parlant de [de la compilation de PHP à partir des sources sur Windows](#) .

10.3.6 Où puis-je trouver le fichier Browser Capabilities ?

Vous pouvez trouver un fichier browscap.ini sur <http://www.garykeith.com/browsers/downloads.asp> .

10.4 Considérations sur les bases de données

Cette section traite de questions relatives aux relations entre PHP et les bases de données. Oui, PHP peut accéder virtuellement à n'importe quelle base de données disponible aujourd'hui.

10.4.1 J'ai entendu dire qu'il était possible d'accéder à Microsoft SQL Server à partir de PHP. Comment est-ce possible?

Sur une machine Windows, vous pouvez tout simplement utiliser le support ODBC inclus avec le pilote ODBC adéquat.

Sur des machines Unix, vous pouvez utiliser le pilote Sybase-CT pour accéder à Microsoft SQL Server car il est (en grande partie) compatible. Sybase a fourni une [version libre des bibliothèques nécessaires pour Linux](#) . Pour les autres systèmes Unix, vous devez contacter Sybase pour obtenir les bibliothèques adéquates. Jetez aussi un oeil à la réponse à la question suivante.

10.4.2 Puis-je accéder à des bases Microsoft Access

Oui. Vous avez déjà tous les outils nécessaires si vous utilisez uniquement Windows 9x/Me, ou NT/2000, et que vous utilisez ODBC avec les pilotes ODBC pour Microsoft Access.

Si vous utilisez PHP sur une machine Unix et que vous voulez vous connecter à une base Access sur une machine Windows, vous aurez besoin des drivers ODBC Unix. [OpenLink Software](#) fournit des pilotes ODBC pour Unix qui peuvent le faire. Il y a un système de pilote libre ou vous pouvez télécharger une version d'évaluation non limitée dans le temps et une version commerciale avec

support à partir de 675\$.

Une autre solution consiste à utiliser un serveur SQL qui a des pilotes ODBC Windows et l'utiliser pour stocker les données, que vous pouvez utiliser à partir de Microsoft Access (en utilisant ODBC) et PHP (en utilisant les pilotes inclus), ou bien utiliser un format de fichier intermédiaire que Access et PHP peuvent traiter tous les deux, comme des fichiers bruts ou des bases de données dBase. A ce sujet, Tim Hayes de Openlink software écrit :

Utiliser une autre base de données comme intermédiaire n'est pas une bonne idée lorsque vous pouvez utiliser ODBC de PHP directement vers vos bases de données - par exemple avec les pilotes Openlink. Si vous avez besoin d'un format de fichier intermédiaire, Openlink a publié Virtuoso (un moteur de base de données virtuel) pour NT, Linux et d'autres plates-formes Unix. Visitez notre [site](#) pour un téléchargement gratuit.

Une solution qui a fait ses preuves est d'utiliser MySQL et ses drivers ODBC sous Windows et de synchroniser les bases de données. Steve Lawrence écrit :

- Installez MySQL sur votre plate-forme conformément aux instructions de MySQL. Vous pouvez l'obtenir sur www.mysql.com (utilisez un miroir!). Aucune configuration particulière n'est nécessaire, mis à part que lorsque vous configurez une base de données et un compte utilisateur, il faut spécifier % dans le champ host, ou bien le nom de la machine Windows avec laquelle vous voulez accéder à MySQL. Notez bien votre nom de serveur, d'utilisateur et votre mot de passe.
- Téléchargez les pilotes MyODBC pour Windows à partir du site de MySQL. La dernière version est myodbc-2_50_19-win95.zip (la version NT est aussi disponible, de même que le code source). Installez les sur votre machine Windows. Vous pouvez tester votre installation avec les utilitaires fournis avec les pilotes.
- Créez un utilisateur ou une source de données système dans votre administration ODBC, dans le panneau de configuration. Donnez un nom de source de données dsn, entrez votre nom d'hôte, identifiant, mot de passe, port, etc pour la base de données configurée à l'étape 1.
- Installez Access avec une installation complète pour être sûr d'avoir tous les composants nécessaires... Vous aurez besoin d'au moins le support ODBC et le gestionnaire de tables liées.
- Maintenant, la partie amusante! Créez un accès à une base de données. Dans la fenêtre de la table, cliquez droit et sélectionnez Lier les Tables, ou, dans le menu Fichier, sélectionnez Obtenir des données externes et alors Lier les Tables. Quand la fenêtre de gestion de fichiers apparaît, sélectionnez les fichiers de type : ODBC. Sélectionnez dsn Système et le nom de votre dsn créée à l'étape 3. Sélectionnez la table à lier, cliquez sur OK, et voilà! Vous pouvez maintenant ouvrir la table et ajouter/supprimer/éditer des données sur votre serveur MySQL! Vous pouvez ainsi exécuter des requêtes, importer/exporter des tables vers MySQL, construire des formulaires et des rapports, etc.

Trucs et astuces :

- Vous pouvez construire vos tables dans Access et les exporter dans MySQL, puis les lier de nouveau. Cela rend la création de table très rapide.
- Lorsque vous créez des tables dans Access, vous devez avoir une clé primaire définie pour avoir accès en écriture à la table. Assurez vous que vous avez bien créé une clé primaire dans MySQL avant de lier le tout à Access
- Si vous changez une table dans MySQL, vous devez la lier de nouveau à Access. Allez dans Outils>suppléments>gestionnaire de tables liées, cherchez votre DSN ODBC, et sélectionnez la table à lier de nouveau. Vous pouvez aussi changer votre source dsn à partir

de là, en cliquant sur l'option toujours demander pour un nouvel emplacement avant de presser OK.

10.4.3 J'ai mis à jour vers PHP 4, et maintenant MySQL me dit toujours "Warning: MySQL: Unable to save result set in ...". Que se passe-t-il?

Il est quasiment sûr que PHP 4 a été compilé avec l'option `--with-mysql` sans donner le chemin vers MySQL. Cela signifie que PHP utilise sa bibliothèque cliente MySQL incluse. Si votre système exécute des applications qui utilisent d'autres versions de clients MySQL, comme PHP 3 en module Apache concurrent, ou `auth-mysql`, il y a un conflit entre les deux versions de ces bibliothèques clientes.

Recompiler PHP 4 et ajouter le chemin vers MySQL avec l'option '`--with-mysql=/votre/chemin/vers/mysql`' résoud généralement le problème.

10.4.4 PHP 5 n'inclut plus les bibliothèques clientes MySQL, qu'est ce que ça implique pour moi? Puis-je toujours utiliser MySQL avec PHP? J'essaie d'utiliser MySQL, mais j'obtiens des erreurs "undefined function".

Oui. PHP supportera toujours MySQL, d'une façon ou d'une autre. Le seul changement avec PHP 5 est que nous ne compilons plus la bibliothèque cliente en elle-même. En voici en vrac quelques justifications :

- La plupart des systèmes incluent maintenant les bibliothèques clientes.
- Ainsi, avoir plusieurs versions des bibliothèques peut être gênant. Par exemple, si vous liez `mod_auth_mysql` à une version et PHP à une autre, et que vous les activez tous les deux dans Apache, vous obtenez un joli plantage. Ainsi, la bibliothèque cliente incluse avec PHP ne fonctionnait pas toujours bien. Le symptôme le plus gênant était que le chemin vers le socket Unix `mysql.sock` n'était pas le bon.
- La maintenance en était lourde et devenait de plus en plus au fur et à mesure des versions.
- Les versions futures de la bibliothèque seront publiées sous licence GPL et nous ne pouvons pas inclure une telle bibliothèque car sa licence n'est pas compatible avec un projet sous licence de type BSD/Apache. Un tel changement dans PHP 5 semble être la meilleure solution.

Ceci n'affectera pas tant que ça les utilisateurs. Les utilisateurs d'Unix, au moins ceux qui savent ce qu'ils font, ont tendance à compiler PHP avec les bibliothèques déjà installées sur leur système, simplement en spécifiant `--with-mysql=/usr` lors de la configuration de PHP. Les utilisateurs de Windows peuvent activer l'extension `php_mysql.dll` dans leur `php.ini`. Pour plus de détails, lisez la [référence MySQL](#) sur les instructions d'installation. Assurez-vous également que `libmysql.dll` est disponible dans le PATH du système. Pour plus de détails sur cela, lisez la FAQ sur [la configuration du PATH sur les systèmes Windows](#). Comme `libmysql.dll` (et plusieurs fichiers relatifs à PHP) existe dans le dossier de PHP, il est recommandé d'ajouter le dossier de PHP à votre PATH système.

10.4.5 Après avoir ajouté le support partagé de MySQL, Apache plante dès que libphp4.so est chargé. Comment résoudre ce problème?

Ceci arrive quand vos bibliothèques MySQL sont liées à pthreads. Vérifiez que vous utilisez ldd. Sinon, téléchargez les sources de MySQL et compilez les ou bien refaites un paquet RPM à partir des sources en enlevant l'option qui active le client threadé dans le fichier de spec. L'une ou l'autre de ces solutions corrigera le problème. Recompiliez alors PHP avec les nouvelles bibliothèques MySQL.

10.4.6 Pourquoi est ce que j'obtiens une erreur comme celle-ci : "Warning: 0 is not a MySQL result index in <file> on line <x>" ou "Warning: Supplied argument is not a valid MySQL result resource in <file> on line <x>?"

Vous essayez d'utiliser un identifiant de résultat qui vaut 0. Le 0 indique que votre requête a échoué pour une quelconque raison. Vous devez alors chercher les erreurs après avoir exécuté une requête et avant même de vouloir traiter le résultat. Une façon propre de le faire est de coder comme ceci :

```
<?php
$result = mysql_query("SELECT * FROM tables_priv");
if (!$result) {
    echo mysql_error();
    exit;
}
?>
```

ou

```
<?php
$result = mysql_query("SELECT * FROM tables_priv")
    or die("Bad query: " . mysql_error());
?>
```

10.5 FAQ sur l'installation

Cette section traite de questions courantes sur la façon d'installer PHP. PHP est disponible sur la plupart des systèmes d'exploitation (sauf les versions de MacOS antérieures à Mac OSX) et pour quasiment n'importe quel serveur web.

Pour installer PHP, suivez les instructions présentes dans le fichier [INSTALL](#) livré avec les sources. Les utilisateurs de Windows doivent aussi lire le fichier [install.txt](#) . Des astuces intéressantes pour les utilisateurs de Windows sont aussi présentes [ici](#) .

10.5.1 Pourquoi ne doit-on pas utiliser Apache 2 dans un environnement threadé multi-processeur de production ?

PHP est un mortier. C'est un mortier utilisé pour construire de belles applications web en utilisant beaucoup de bibliothèques ensemble, apparaissant comme une seule entité à travers un langage intuitif et facile à apprendre. La flexibilité et la puissance de PHP se fondent sur la stabilité et la robustesse de la plate-forme fondamentale. Il a besoin d'un OS qui fonctionne, d'un serveur web qui fonctionne et de bibliothèques externes pour coller le tout. Lorsqu'un seul de ces éléments arrête subitement de fonctionner, PHP doit identifier le problème et le réparer au plus vite. En rendant le cadre fondamental plus complexe en ne séparant pas les exécutions des threads, ni les segments mémoires, ni un endroit clos pour traiter chaque requête entrante, des pieds d'argile sont introduits dans le système PHP.

Et finalement, cette mise en garde contre les environnements MPM threadés n'est pas aussi forte pour les environnements Windows, où les bibliothèques sont mieux threadées.

10.5.2 Unix/Windows : où doit être placé mon fichier

Par défaut sous Unix, il doit être placé dans `/usr/local/lib` qui est en fait `<install-path>/lib`. La plupart des personnes voudront changer ceci lors de la compilation avec l'option `--with-config-file-path`. Vous pouvez par exemple le régler de cette façon :

```
--with-config-file-path=/etc
```

Et alors vous copierez le fichier `php.ini-dist` livré avec les sources vers `/etc/php.ini` et l'éditer pour l'adapter à vos besoins.

```
--with-config-file-scan-dir=PATH
```

Sous Windows, le chemin par défaut de `php.ini` est le répertoire de Windows. Si vous utilisez le serveur web Apache, `php.ini` est tout d'abord cherché dans le répertoire d'installation de Apache, c'est à dire `c:\program files\apache group\apache`. De cette façon, vous pouvez avoir un `php.ini` différent pour chaque version de Apache installée.

Consultez aussi le chapitre sur le [fichier de configuration](#).

10.5.3 Unix : j'ai installé PHP, mais à chaque fois que je charge un document, j'obtiens l'erreur 'Document Contains No Data'! Que se passe-t-il?

Cela signifie probablement que PHP rencontre un problème et génère un fichier core. Consultez vos fichiers de logs de votre serveur pour voir si c'est le cas, et tentez de reproduire le problème avec un test simple. Si vous savez utiliser 'gdb', il serait très utile de fournir un backtrace avec votre rapport de bogue, afin d'aider les développeurs à cerner le problème. Si vous utilisez PHP en module Apache, essayez ceci :

- Stoppez vos processus httpd
- `gdb httpd`
- Stoppez vos processus httpd
- `> run -X -f /path/to/httpd.conf`
- Pointez alors avec votre navigateur vers l'URL posant problème.
- `> run -X -f /path/to/httpd.conf`
- Si vous obtenez un fichier core, gdb doit alors vous en informer.
- tapez : `bt`

- Vous devriez inclure votre backtrace dans votre rapport de bogue. Celui-ci doit être posté sur <http://bugs.php.net/> .

Si votre script utilise les expressions rationnelles (`ereg` et consorts), assurez-vous que PHP et Apache ont été compilés avec les mêmes outils d'expression rationnelles. Cela doit être automatiquement le cas avec PHP et Apache 1.3.x.

10.5.4 Unix : J'ai installé PHP en utilisant des fichiers sources RPM, mais apache ne traite pas les pages PHP. Que se passe-t-il?

En supposant que vous avez installé à la fois Apache et PHP à partir de fichiers RPM, vous devrez commenter ou ajouter au moins quelques unes des lignes suivantes dans votre fichier `httpd.conf` :

```
# Extra Modules
AddModule mod_php.c
AddModule mod_php3.c
AddModule mod_perl.c

# Extra Modules
LoadModule php_module          modules/mod_php.so
LoadModule php3_module         modules/libphp3.so      # for PHP 3
LoadModule php4_module         modules/libphp4.so      # for PHP 4
LoadModule perl_module         modules/libperl.so
```

Et ajouter :

```
AddType application/x-httpd-php3 .php3      # for PHP 3
AddType application/x-httpd-php .php        # for PHP 4
```

... aux propriétés globales ou aux propriétés du `VirtualDomain` ou vous voulez que PHP officie.

10.5.5 Unix : J'ai installé PHP 3 en utilisant des paquets source RPM, mais ils ne compilent pas avec le support de la base de données dont j'ai besoin! Que se passe-t-il?

La façon dont PHP 3 est compilé rend la construction flexible de paquets RPM très peu aisée. Ceci est résolu avec PHP 4. Pour PHP 3, nous vous conseillons de suivre le mécanisme décrit dans le fichier `INSTALL.REDHAT` inclus avec les sources. Si vous persistez à vouloir construire des fichiers RPM de PHP 3, lisez ceci...

Les mainteneurs des paquets RPM ont configuré les paquets RPM à s'installer sans le support des bases de données pour simplifier l'installation et parce que les paquets RPM utilisent le répertoire `/usr` au lieu du répertoire standard `/usr/local/` pour les fichiers. Vous devez spécifier au fichier de spécification du paquet RPM quelle base de données vous voulez utiliser et quel est chemin complet du serveur de votre base de données.

L'exemple ci-dessous explique la procédure pour ajouter le support de la base de données populaire, MySQL, en utilisant une installation en module d'Apache.

Bien sûr, toutes ces informations peuvent être ajustées pour n'importe quel serveur de bases de données que PHP supporte. Nous supposons que vous avez installé MySQL et Apache complètement avec des paquets RPM pour cet exemple.

- Premièrement, supprimer mod_php3 :

```
rpm -e mod_php3
```

- Ensuite, récupérer le paquet RPM et installez-le, SANS --rebuild

```
rpm -Uvh mod_php3-3.0.5-2.src.rpm
```

- Ensuite, éditez le fichier /usr/src/redhat/SPECS/mod_php3.spec

Dans la section %build, ajoutez le support de la base de données désiré ainsi que son chemin.

Pour MySQL, vous devez ajouter --with-mysql=/usr La section %build devrait ressembler à quelque chose comme ceci :

```
./configure --prefix=/usr \  
--with-apxs=/usr/sbin/apxs \  
--with-config-file-path=/usr/lib \  
--enable-debug=no \  
--enable-safe-mode \  
--with-exec-dir=/usr/bin \  
--with-mysql=/usr \  
--with-system-regex
```

- Une fois cette modification effectuée, compiler le binaire RPM comme ceci :

```
rpm -bb /usr/src/redhat/SPECS/mod_php3.spec
```

- Ensuite, installez le RPM

```
rpm -ivh /usr/src/redhat/RPMS/i386/mod_php3-3.0.5-2.i386.rpm
```

Assurez-vous de redémarrer Apache et que vous avez PHP 3 avec le support MySQL en utilisant le paquet RPM. Notez qu'il est probablement plus facile de compiler directement depuis l'archive tarball PHP 3 et de suivre les instructions du fichier INSTALL.REDHAT dans les sources de la distribution.

10.5.6 Unix : J'ai patché Apache avec l'extension FrontPage et subitement, PHP ne fonctionne plus. Est-ce que PHP est incompatible avec l'extension FrontPage pour Apache ?

Non, PHP fonctionne très bien avec l'extension FrontPage. Le souci vient du fait que le patch pour l'installation de FrontPage modifie la structure d'Apache, qui est relié à PHP. Recompilez PHP (en utilisant 'make clean ; make') après avoir patché Apache avec l'extension FrontPage devrait résoudre ce problème.

10.5.7 Unix/Windows : J'ai installé PHP, mais lorsque je tente d'accéder à un fichier contenant un script PHP via mon navigateur, j'obtiens un écran vide.

Affichez le code source du document dans votre navigateur et vous devriez probablement trouver le code source de votre script PHP. Cela signifie que le serveur web n'a pas envoyé le script à PHP pour interprétation. Quelque chose est donc incorrecte dans le fichier de configuration de votre serveur web - re-vérifier la configuration du serveur web en vous référant aux instructions d'installations de PHP.

10.5.8 Unix/Windows : J'ai installé PHP, mais lorsque je tente d'accéder à un fichier contenant un script PHP via mon navigateur, j'obtiens une erreur de type 'server 500 error'.

Quelque chose se passe mal lorsque le serveur tente d'utiliser PHP. Pour tenter de récupérer un message d'erreur, depuis la ligne de commande, placez-vous dans le répertoire contenant l'exécutable PHP (php.exe sous Windows) et exécutez la commande `php -i` . Si PHP a un problème quelconque l'empêchant de fonctionner, un message d'erreur devrait s'afficher qui devrait expliquer comment résoudre ce souci. Si un écran de code HTML apparaît (la sortie de la fonction `phpinfo`), cela signifie que PHP fonctionne correctement et que le problème doit certainement venir de la configuration de votre serveur web que vous devriez re-vérifier.

10.5.9 Quelques systèmes d'exploitations : J'ai installé PHP sans erreur, mais lorsque je tente de démarrer Apache, j'obtiens une erreur du type '

Cela n'a actuellement rien à voir avec PHP mais avec la bibliothèque cliente MySQL. Suivant les versions, elle a besoin que PHP soit compilé avec l'option `--with-zlib` , d'autre non. Ce problème est également traité dans la FAQ de MySQL.

10.5.10 Windows : J'ai installé PHP, mais lorsque j'accède à un fichier contenant un script PHP via mon navigateur, j'obtiens l'erreur :

Ce message d'erreur signifie que PHP a échoué lors de l'affichage. Pour tenter de récupérer un message d'erreur, depuis la ligne de commande, placez-vous dans le répertoire contenant l'exécutable PHP (php.exe sous Windows) et exécutez la commande `php -i` . Si PHP a un quelconque souci de fonctionnement, alors un message d'erreur le décrivant s'affichera. Si vous obtenez un écran de code HTML (le contenu du résultat de la fonction `phpinfo`), alors PHP fonctionne correctement.

Si PHP fonctionne depuis la ligne de commande, tentez d'accéder à votre script encore une fois via votre navigateur. S'il échoue toujours, alors, il se peut que ce soit l'un des soucis suivants :

- Les permissions de votre script PHP des fichiers `php.exe` , `php4ts.dll` , `php.ini` ou de toute extension nécessaire à PHP que vous tentez de charger, sont telles que l'utilisateur internet anonyme de votre système ISUR_ <machinename> ne peut pas y accéder.
- le script PHP n'existe pas (ou n'est pas à l'endroit que vous pensez, relativement au répertoire racine de votre serveur web). Notez que pour le serveur web IIS, vous pouvez vérifier cela en cochant la case 'vérifier si le fichier existe' lors de la configuration de l'exécution des scripts dans le gestionnaire de services Internet. Si un fichier de script n'existe pas, le serveur web retournera une erreur 404. IIS a également l'avantage d'effectuer toutes les identifications requises à votre place, basés sur les permissions NT LanMan, sur votre fichier de script.

10.5.11 Windows : J'ai suivi toutes les instructions, mais je n'arrive toujours pas à faire fonctionner PHP et IIS ensemble !

Assurez-vous que chaque utilisateur qui a besoin d'exécuter un script PHP possède les droits requis pour exécuter le fichier `php.exe` ! IIS utilise un utilisateur anonyme qui est ajouté lors de l'installation de IIS. Cet utilisateur doit avoir les droits suffisant sur le fichier `php.exe` . De même,

tous les utilisateurs enregistrés doivent posséder les droits requis pour exécuter le fichier php.exe .
Pour IIS4, vous devez lui dire que PHP est un moteur de script. De plus, vous devriez lire cette [FAQ](#) .

10.5.12 Lorsque vous exécutez PHP comme CGI avec IIS, PWS, OmniHTTPD ou Xitami, j'obtiens l'erreur suivante :

Vous devez définir la directive `cgi.force_redirect` à 0 . Par défaut, elle vaut 1 , donc, soyez sûr que cette directive n'est pas commentée (précédé d'un point virgule). Comme toutes les directives, elles sont définies dans le php.ini .

Comme la valeur par défaut vaut 1 , il est impératif que vous soyez sûrs à 100% que le bon fichier php.ini a été lu. Lisez cette [FAQ](#) pour plus de détails.

10.5.13 Comment puis-je savoir si mon

Pour être sûr que votre php.ini a été lu par PHP, effectuez un appel à la fonction `phpinfo` et vers le haut du document résultant, il devrait figurer une liste appelée Configuration File (php.ini) . Cela vous indiquera où PHP a cherché le php.ini et si oui ou non il l'a lu. S'il n'apparaît qu'un chemin vers un répertoire existant, cela signifie que vous devez copier votre php.ini dans ce répertoire. Si le php.ini est présent dans le chemin, cela signifie qu'il a bien été lu.

Si le php.ini a bien été lu et que vous exécutez PHP comme module, alors assurez-vous de redémarrer le serveur web après avoir effectué les modifications à votre php.ini .

10.5.14 Où dois-je ajouter mon répertoire PHP à la variable

Sous Windows NT, 2000, XP et 2003 :

- Allez dans le centre de contrôle et ouvrez l'icône système (Démarrer -> Paramètres -> Panneau de configuration -> Système ou juste Démarrer -> centre de contrôle -> Système pour Windows XP/2003)
- Allez à l'onglet "Avancé"
- Cliquez sur le bouton "Variables d'environnements"
- Regardez dans le panneau "Variables systèmes"
- Trouvez l'entrée Path (vous devriez avoir à faire descendre l'ascenseur pour le trouver)
- Double cliquez sur l'entrée Path
- Entrez votre répertoire PHP à la fin, sans oublier le point virgule (;) avant (e.g. ;C:\php)
- Confirmez en cliquant sur OK et redémarrer votre ordinateur

Sous Windows 98/Me, vous devez éditer le fichier autoexec.bat :

- Ouvrez Notepad (Démarrer -> Exécuter et entrez notepad)
- Ouvrez le fichier C:\autoexec.bat
- Ajoutez une nouvelle ligne à la fin du fichier : `set PHPRC=C:\php` (remplacez C:\php par le dossier où se trouve le php.ini). Notez que le chemin ne doit pas contenir d'espaces. Par exemple, si vous avez installé PHP dans le dossier C:\Program Files\PHP , vous devez indiquer le chemin sous la forme C:\PROGRA~1\PHP .
- Sauvegardez le fichier et redémarrer l'ordinateur

Note

Assurez-vous de redémarrer l'ordinateur après avoir suivi cette procédure afin que les modifications sur la variable PATH soient bien prises en compte.

Le manuel PHP vous encourage à copier les fichiers dans le dossier système de Windows car ce dossier (C:\Windows , C:\WINNT , etc.) se trouve par défaut dans le PATH système. Copier les fichiers dans le dossier système de Windows est depuis déprécié et peut causer des problèmes.

10.5.15 Comment rendre le fichier

Il y a plusieurs façons de faire cela. Si vous utilisez Apache, lisez leurs instructions spécifiques d'installation ([Apache 1](#) , [Apache 2](#)) sinon, vous devez définir la variable d'environnement PHPRC :

Sous Windows NT, 2000, XP et 2003:

- Allez dans le centre de contrôle et ouvrez l'icône système (Démarrer -> Paramètres -> Panneau de configuration -> Système ou juste Démarrer -> centre de contrôle -> Système pour Windows XP/2003)
- Allez à l'onglet "Avancé"
- Cliquez sur le bouton "Variables d'environnements"
- Regardez dans le panneau "Variables systèmes"
- Cliquez sur "Nouveau" et entrez "PHPRC" comme nom de variable et le dossier où se trouve votre fichier php.ini comme valeur (e.g. C:\php)
- Confirmez en cliquant sur OK et redémarrer votre ordinateur

Sous Windows 98/Me, vous devez éditer le fichier autoexec.bat :

- Ouvrez Notepad (Démarrer ->Exécuter et entrez notepad)
- Ouvrez le fichier C:\autoexec.bat
- Ajouter une nouvelle ligne à la fin du fichier : set PHPRC C:\php (remplacez C:\php avec le dossier où se trouve le fichier php.ini)
- Sauvegardez le fichier et redémarrer l'ordinateur

10.5.16 Est-il possible d'utiliser la négociation sur le contenu fournie par Apache (option MultiViews) avec PHP ?

Si les liens vers les fichiers PHP incluent l'extension, tout fonctionne parfaitement. Cette entrée de la FAQ traite uniquement du cas où les liens vers les fichiers PHP n'incluent pas l'extension et que vous voulez utiliser la négociation sur le contenu fourni par Apache pour choisir les fichiers PHP depuis une URL qui ne contient pas d'extension. Dans ce cas, remplacez la ligne AddType application/x-httpd-php .php par :

```
# PHP 4
AddHandler php-script php
AddType text/html php

# PHP 5
AddHandler php5-script php
AddType text/html php
```

Cette solution ne fonctionne pas pour Apache 1 car le module PHP ne captura pas le php-script .

10.5.17 Est-ce que PHP est limité au traitement des méthodes GET et POST ?

Non, il est possible de gérer tout type de méthode, comme CONNECT. Les bons entêtes de réponse peuvent être envoyés avec la fonction `header`. Si seules les méthodes POST et GET doivent être gérées, vous pouvez configurer Apache comme ce qui suit :

```
<LimitExcept GET POST>
Deny from all
</LimitExcept>
```

10.6 Problèmes de compilation

Cette section couvre les erreurs les plus communes pouvant se produire lors de la compilation de PHP.

10.6.1 J'ai téléchargé la dernière version des sources de PHP en utilisant CVS, mais il n'y a pas de script configure!

Vous devez avoir le logiciel GNU autoconf d'installé pour générer le script configure à partir de `configure.in`. Lancez juste `./buildconf` à la racine du répertoire des sources après avoir récupéré celles-ci à partir du serveur CVS. (De plus, à moins que vous ne lanciez `configure` avec l'option `--enable-maintainer-mode`, le script configure ne sera pas automatiquement reconstruit si `configure.in` est mis à jour, vous obligeant à le faire à la main quand vous remarquez que `configure.in` est mis à jour. Une conséquence de ceci est que l'on trouve des choses telles que dans votre `Makefile` après que `configure` ou `config.status` soit lancé.)

10.6.2 J'ai des problèmes pour configurer PHP avec Apache. On m'indique que

Vous devez spécifier au script de configuration (`configure`) l'emplacement du répertoire où sont les sources de Apache. Cela signifie que vous devez spécifier `--with-apache=/chemin/vers/apache` et **pas** `--with-apache=/chemin/vers/apache/src`.

10.6.3 Pendant la configuration de PHP (

Assurez-vous de bien avoir lu les [instructions](#) d'installation et d'avoir `flex` et `bison` d'installés pour compiler PHP. Selon votre système, vous devrez installer `bison` et `flex` à partir de sources ou bien de paquets, tel qu'un RPM.

10.6.4 Quand je lance Apache, j'obtiens le message suivant :

Cette erreur survient généralement quand quelqu'un compile le coeur Apache comme bibliothèque partagée DSO. Essayez de reconfigurer Apache, en vous assurant d'utiliser les options suivantes :

```
--enable-shared=max --enable-rule=SHARED_CORE
```

Pour davantage d'information, lisez le fichier `INSTALL` à la racine de votre répertoire source Apache ou bien [le manuel des bibliothèques DSO](#) .

10.6.5 Quand je lance le `./configure`, on me dit que les fichiers d'en-tête de GD, gdbm, ... ne sont pas trouvés!

Vous pouvez forcer le script `configure` à chercher les fichiers d'en-tête à des endroits non-standard en passant des options supplémentaires au préprocesseur C et au `lieur`, par exemple :

```
CPPFLAGS=-I/path/to/include LDFLAGS=-L/path/to/library ./configure
```

Si vous utilisez une variante de `csh`, utilisez plutôt :

```
env CPPFLAGS=-I/path/to/include LDFLAGS=-L/path/to/library ./configure
```

10.6.6 Quand le fichier

Vous devez mettre à jour votre version de `bison`. Vous pourrez trouver la dernière version sur <http://www.gnu.org/software/bison/bison.html> .

10.6.7 Quand je lance

Certaines anciennes versions de `make` ne déplacent pas correctement les versions compilées des fichiers dans le répertoire `functions`. Essayez de lancer `cp *.o functions` et de relancer `make` pour voir si le problème est résolu. Si tel est le cas, vous devriez vraiment mettre à jour votre version de GNU `make`.

10.6.8 Au moment de lier PHP, il y a des références indéfinies.

Jetez un oeil à la ligne de lien et assurez-vous que toutes les bibliothèques nécessaires ont été incluses à la fin. Celles qui manquent probablement sont `-ldl` et les bibliothèques relatives aux bases de données dont vous voulez le support.

Si vous compilez par rapport à Apache 1.2.x, souvenez-vous de bien mettre les informations appropriées dans la ligne `EXTRA_LIBS` du fichier `CONfiguration` et relancez le script `CONfigure` de Apache. Lisez le fichier `INSTALL` distribué avec pour davantage d'informations.

Des personnes nous ont rapporté qu'elle devait ajouter `-ldl` immédiatement après `libphp4.a` lors de la compilation avec Apache.

10.6.9 Je ne vois pas comment compiler PHP avec Apache 1.3.

C'est relativement simple. Suivez attentivement ces étapes :

- Téléchargez la dernière version de Apache 1.3 sur <http://www.apache.org/dist/httpd/> .
- Désarchivez le tout, par exemple dans `/usr/local/src/apache-1.3` .
- Compilez PHP en lançant tout d'abord `./configure --with-apache=/<path>/apache-1.3` (remplacez `<path>` par le chemin de votre répertoire `apache-1.3`)

- Tapez `make` puis `make install` pour compiler PHP et copier les fichiers nécessaires dans le répertoire source Apache.
- Déplacez-vous dans votre répertoire `/<path>/apache-1.3/src` et éditez le fichier `Configuration`. Ajoutez y : `AddModule modules/php4/libphp4.a`.
- Tapez : `./configure` puis `make`.
- Vous devriez alors avoir une bibliothèque `httpd` avec le support de PHP!

Note: Vous pouvez utiliser le nouveau script `./configure` de Apache. Suivez les instructions du fichier `README.configure` fourni avec votre version de Apache. Lisez aussi le fichier `INSTALL` inclus avec les sources de PHP.

10.6.10 j'ai suivi toutes les étapes pour installer le module Apache sous Unix, mais malgré tout, mes scripts PHP s'affichent en clair dans mon navigateur ou celui-ci me demande de sauver le fichier.

Cela signifie que le module PHP n'est pas chargé, pour une raison ou pour une autre. Avant de chercher de l'aide ailleurs, veuillez vérifier ces quelques points :

- Assurez-vous que le binaire `httpd` que vous exécutez est bien le nouveau binaire que vous avez compilé. Pour cela, essayez de lancer : `:/chemin/vers/le/binaire/httpd -l`
- Si vous ne voyez pas `mod_php4.c` dans la liste, c'est que vous n'utilisez pas le bon binaire. Trouvez et installez correctement le bon binaire.
- Assurez-vous que vous avez bien ajouté le bon type `Mime` à un de vos fichiers Apache `.conf`. Ce devrait être : `AddType application/x-httpd-php3 .php3` (pour PHP 3)

ou `AddType application/x-httpd-php .php` (pour PHP 4)

Assurez-vous aussi que cette ligne `Addtype` n'est pas dissimulée dans un contexte de `<Virtualhost>` ou `<Directory>` qui l'empêcherait de s'appliquer à l'emplacement de vos scripts.

- Enfin, l'emplacement par défaut des fichiers de configuration Apache a changé entre Apache 1.2 et Apache 1.3. Vous devriez vérifier que les fichiers de configuration auxquels vous ajoutez la ligne `Addtype` sont bien ceux qui sont pris en compte. Vous pouvez introduire une erreur de syntaxe dans votre `httpd.conf` (ou bien tout autre changement incorrect) pour vous assurer que c'est bien ce fichier qui est pris en compte.

10.6.11 Il est dit d'utiliser

Notez que le fichier `libphp4.a` n'est pas supposé exister. Le processus `apache` le créera!

10.6.12 Quand j'essaie de compiler Apache avec PHP en module statique en utilisant

C'est un mauvais message d'erreur de Apache qui n'apparait plus dans des versions plus récentes.

10.6.13 Quand j'essaie de compiler PHP avec

Il y a trois choses à vérifier ici. Tout d'abord, quand Apache crée le script Perl apxs, il s'interrompt parfois en étant compilé sans le bon compilateur ou les bonnes options. Trouvez votre script apxs (lancez la commande `which apxs`), qui se trouve souvent à `/usr/local/apache/bin/apxs` ou bien `/usr/sbin/apxs`. Editez le et vérifiez que des lignes similaires sont présentes :

```
my $CFG_CFLAGS_SHLIB = ' ';          # substituted via Makefile.tmpl
my $CFG_LD_SHLIB     = ' ';          # substituted via Makefile.tmpl
my $CFG_LDFLAGS_SHLIB = ' ';        # substituted via Makefile.tmpl
```

Si c'est ce que vous voyez, vous avez trouvé votre problème. Elles peuvent contenir juste des espaces ou d'autres valeurs incorrectes, comme 'q()'. Changez ces lignes pour obtenir :

```
my $CFG_CFLAGS_SHLIB = '-fpic -DSHARED_MODULE'; # substituted via Makefile.tmpl
my $CFG_LD_SHLIB     = 'gcc';                  # substituted via Makefile.tmpl
my $CFG_LDFLAGS_SHLIB = q(-shared);           # substituted via Makefile.tmpl
```

Le deuxième problème potentiel est uniquement relatif aux distributions Red Hat 6.1 et 6.2. The scripts apxs de Red Hat est défectueux. Cherchez cette ligne :

```
my $CFG_LIBEXECDIR = 'modules';          # substituted via APACI install
```

Si vous la voyez telle quelle, changez la en :

```
my $CFG_LIBEXECDIR = '/usr/lib/apache'; # substituted via APACI install
```

Enfin, si vous reconfigurez/réinstallez Apache, lancez un `make clean` entre votre `./configure` et votre `make`.

10.6.14 Pendant le

Pendant le `make`, si vous rencontrez des problèmes identiques à celui-ci :

```
microtime.c: In function `php_if_getrusage':
microtime.c:94: storage size of `usg' isn't known
microtime.c:97: `RUSAGE_SELF' undeclared (first use in this function)
microtime.c:97: (Each undeclared identifier is reported only once
microtime.c:97: for each function it appears in.)
microtime.c:103: `RUSAGE_CHILDREN' undeclared (first use in this function)
make[3]: *** [microtime.lo] Error 1
make[3]: Leaving directory `/home/master/php-4.0.1/ext/standard'
make[2]: *** [all-recursive] Error 1
make[2]: Leaving directory `/home/master/php-4.0.1/ext/standard'
make[1]: *** [all-recursive] Error 1
make[1]: Leaving directory `/home/master/php-4.0.1/ext'
make: *** [all-recursive] Error 1
```

Votre système est défectueux. Vous devez corriger vos fichiers `/usr/include` en installant un paquet `glibc-devel` qui correspond à votre version de la `glibc`. Cela n'a rien à voir avec PHP. Pour vous en convaincre, essayez ceci :

```
$ cat >test.c <<X
#include <sys/resource.h>
X
$ gcc -E test.c >/dev/null
```

Si vous obtenez des erreurs, c'est que vos fichiers d'en-tête sont mauvais.

10.6.15 Quand je compile PHP avec le support MySQL, le configure se passe bien, mais pendant le

Tout d'abord, il est important de savoir que ce n'est qu'un Warning et pas une erreur fatale. Comme c'est souvent la dernière erreur vu lors du make , ça a l'air d'une erreur fatale, mais ça n'en est pas une. Bien sûr, si vous demandez à votre compilateur de stopper à chaque Warning, ça en deviendra une. Notez aussi que le support de MySQL est activé par défaut.

Note
Depuis PHP 4.3.2, vous verrez le texte suivant après la compilation (make) :
Build complete. (It is safe to ignore warnings about tempnam and tmpnam).

10.6.16 Je veux mettre à jour mon PHP. Où puis-je trouver la ligne

vous pouvez jeter un oeil au fichier config.nice dans votre répertoire source ou sinon simplement exécuter un script

```
<?php phpinfo(); ?>
```

Au début du résultat affiché figure la ligne ./configure qui fût utilisée lors de la configuration de votre PHP actuel.

10.6.17 Quand je compile PHP avec le support de la bibliothèque GD, j'obtiens des erreurs de compilation étrange, voire même des erreurs de segmentation.

Assurez-vous que votre bibliothèque GD et PHP sont liés aux mêmes bibliothèques (libpng, par exemple).

10.6.18 Quand je compile PHP, j'obtiens des erreurs aléatoires, voire même tout s'arrête. J'utilise Solaris.

L'utilisation d'utilitaires non-GNU pour compiler PHP peut poser problème. Assurez-vous de bien utiliser des outils GNU pour être certain que votre compilation arrive à terme. Par exemple, sous Solaris, utiliser les version SunOS BSD-compatible ou Solaris de sed ne fonctionnera pas, mais utiliser les versions GNU ou Sun POSIX (xpg4) de sed fonctionnera. Liens : [GNU sed](#) , [GNU flex](#) , et [GNU bison](#) .

10.7 Utiliser PHP

Cette section regroupe plusieurs erreurs que vous pouvez rencontrer lors de l'écriture de vos scripts PHP.

10.7.1 J'aimerais écrire un script PHP générique qui pourrait traiter les données provenant de tout formulaire. Comment savoir quelles variables de la méthode POST sont disponibles ?

PHP fournit plusieurs variables pré-définies , comme la super globale \$_POST . Vous pouvez boucler sur \$_POST puisque c'est un tableau associatif de toutes les valeurs POSTées. Par exemple, bouclons dessus simplement avec foreach , vérifions les valeurs vides et affichons les.

```
<?php
$empty = $post = array();
foreach ($_POST as $nomvar => $valeurvar) {
    if (empty($valeurvar)) {
        $empty[$nomvar] = $valeurvar;
    } else {
        $post[$nomvar] = $valeurvar;
    }
}

echo '<pre>';
if (empty($empty)) {
    print "Aucune valeur POSTée n'est vide, postées :\n";
    var_dump($post);
} else {
    print "Nous avons " . count($empty) . " valeurs vides\n";
    print "Postées :\n"; var_dump($post);
    print "Vides :\n"; var_dump($empty);
    exit;
}
echo '</pre>';
?>
```

Note

Superglobales : disponibilité

Depuis PHP 4.1.0, les tableaux superglobaux tels que \$_GET , \$_POST et \$_SERVER , etc. sont disponibles. Pour plus d'informations, lisez la section superglobaux

10.7.2 Il faut que je convertisse tous les guillemets simples (') en un anti-slash suivi d'un guillemet simple (\'). Comment le faire avec une expression régulière ? J'aimerais aussi convertir " en \" et \ en \\.

La fonction addslashes le fera. Voir aussi mysql_escape_string . Vous pouvez aussi supprimer les anti-slashes avec stripslashes .

Note

Note concernant la directive : magic_quotes_gpc

La directive PHP magic_quotes_gpc est par défaut à on . En bref, elle applique la fonction addslashes sur toutes vos données issues de GET, POST et COOKIE. Vous pouvez utiliser la fonction stripslashes pour supprimer cet effet.

10.7.3 Tous mes " se transforment en \" et mes ' en \', comment me débarrasser de tous ces anti-slashes ? Comment et pourquoi sont-ils apparus ?

La fonction PHP [stripslashes](#) supprimera ces anti-slashes de votre chaîne de caractères. La plupart du temps, ils apparaissent car la directive PHP [magic_quotes_gpc](#) est activée.

Note

Note concernant la directive : [magic_quotes_gpc](#)

La directive PHP [magic_quotes_gpc](#) est par défaut à on . En bref, elle applique la fonction [addslashes](#) sur toutes vos données issues de GET, POST et COOKIE. Vous pouvez utiliser la fonction [stripslashes](#) pour supprimer cet effet.

10.7.4 Quand je fais ce qui suit, l'affichage se fait dans le mauvais ordre

:

Pour pouvoir utiliser le résultat de votre fonction dans une expression (comme le concaténer avec une chaîne comme dans cet exemple), vous devez retourner la valeur avec [return](#) , et non pas l'afficher avec [echo](#) .

10.7.5 Hey, où sont mes nouvelles lignes ?

En PHP, la fin d'un bloc de code est soit ">" ou ">\n" (où \n signifie une nouvelle ligne). Donc dans l'exemple plus haut, les phrases affichées le seront sur une seule ligne, car PHP oublie les nouvelles lignes après la fin du bloc. Cela signifie que vous devez insérer une nouvelle ligne de plus après chaque bloc de code PHP pour la lui faire afficher.

Pourquoi PHP fait-il cela ? Car lors du formatage du HTML, cela vous simplifie la vie car vous ne voulez pas de cette nouvelle ligne, mais vous devez créer de très longues lignes ou rendre la source brute de la page illisible pour arriver à cet effet.

10.7.6 J'obtiens le message 'Warning: Cannot send session cookie - headers already sent...' ou 'Cannot add header information - headers already sent...'

Les fonctions [header](#) , [setcookie](#) , et les [fonctions de session](#) doivent ajouter des en-têtes au flux de sortie, mais ceux-ci ne peuvent être envoyés qu'avant le reste du contenu. Il ne doit y avoir aucun affichage avant d'utiliser ces fonctions, comme le HTML par exemple. La fonction [headers_sent](#) vérifiera si votre script a déjà envoyé des en-têtes. Voyez aussi [les fonctions de bufferisation de sortie](#) .

10.7.7 J'ai besoin d'accéder à des informations dans l'en-tête de requête directement. Comment puis-je le faire ?

La fonction [getallheaders](#) le fera si vous exécutez PHP en tant que module Apache. Le code suivant vous montrera tous les en-têtes de requête :

```
<?php
$headers = getallheaders();
foreach ($headers as $nom => $contenu) {
    echo "headers[$nom] = $contenu<br />\n";
}
?>
```

Voir aussi [apache_lookup_uri](#) , [apache_response_headers](#) et [fsockopen](#) .

10.7.8 Quand j'essaye d'utiliser l'identification avec IIS j'obtiens

Le modèle sécuritaire de IIS est en faute. C'est un problème commun à tous les programmes CGI fonctionnant avec IIS. Une alternative est de créer un fichier HTML (non exécuté par PHP) comme page d'entrée dans le dossier où il faut s'identifier. Utilisez alors une balise META pour rediriger vers la page PHP, ou encore proposez un lien vers celle-ci. PHP reconnaîtra alors l'identification correctement. Avec le module ISAPI, cela n'est pas un problème. Cela ne devrait pas non plus affecter d'autres serveurs NT. Pour plus d'informations, voyez : <http://support.microsoft.com/kb/q160422/> et la section du manuel concernant l' [identification HTTP](#) .

10.7.9 Windows: Je ne peux pas accéder aux fichiers partagés sur un autre ordinateur utilisant IIS

Vous devez modifier le service Go to Internet Information Services . Localisez votre fichier PHP et éditez ces propriétés. Placez-vous sur l'onglet File Security , Edit -< Anonymous access and authentication control .

Vous pouvez résoudre ce souci soit en décochant la case Anonymous Access et en laissant la case Integrated Window Authentication cochée, soit en cochant la case Anonymous Access et en éditant l'utilisateur qui ne doit pas avoir les droits d'accès.

10.7.10 Mon script PHP fonctionne avec IE et Lynx, mais avec Netscape une portion de ce qui devrait s'afficher manque. Quand j'affiche la source HTML de la page, je vois le contenu avec IE mais pas avec Netscape.

Netscape est plus strict que IE concernant les balises HTML (comme les tables). Faire valider votre HTML généré par un validateur HTML, comme validator.w3.org , peut se révéler utile. Par exemple, une `</table>` manquante peut provoquer ce problème.

De plus, IE et Lynx ignorent les NULs (`\0`) dans le flux HTML, Netscape non. La meilleure façon de le vérifier est de compiler la version en [ligne de commande](#) de PHP (aussi connue sous le nom de version CGI) et d'exécuter vos scripts à partir de la console. Sous *nix, redirigez la sortie sûr `od -c` et cherchez les caractères `\0` . Si vous êtes sous Windows, vous devez trouver un éditeur ou un autre programme qui vous permettra de visualiser les fichiers binaires. Lorsque Netscape rencontre un NUL dans un fichier, il n'affichera la plupart du temps rien dans cette ligne, alors que IE et Lynx le feront.

10.7.11 Comment mélanger XML et PHP ? PHP se plaint de mes balises `<?xml !`

Pour inclure `<?xml` dans votre code PHP, vous devrez désactiver les short tags en configurant la directive PHP [short_open_tags](#) à 0 . Vous ne pouvez pas modifier cette directive avec [ini_set](#) . Que [short_open_tags](#) soit à on ou off, vous pouvez toujours faire ceci : `<?php echo '<?xml'; ?>` . La

valeur par défaut pour cette directive est on.

10.7.12 Comment utiliser PHP avec FrontPage ou d'autres éditeurs HTML qui insistent pour mettre mon code ailleurs ?

La façon la plus simple est d'activer l'utilisation des balises ASP dans votre code PHP. Cela vous permettra d'utiliser des délimiteurs de code à la manière de ASP : `<%` et `%>`. Quelques éditeurs HTML populaires gèrent cela plus habilement. Pour activer les balises ASP, vous devez configurer la variable `asp_tags` de `php.ini`, ou utiliser la directive Apache appropriée.

10.7.13 Où puis-je trouver une liste complète des variables pré-définies que je peux utiliser dans mes scripts PHP ?

Lisez la page du manuel qui concerne les [variables pré-définies](#) vu qu'elle présente une liste partielle des variables pré-définies disponibles dans votre script. Une liste complète des variables disponibles (et beaucoup d'informations) peut être vue en appelant la fonction `phpinfo`. Lisez la section du manuel traitant des [variables non-issues de PHP](#), elle décrit des scénarios communs pour les variables externes, comme celles issues d'un formulaire HTML, d'un cookie, et de l'URL.

Note
register_globals : note importante
Depuis PHP 4.2.0, la valeur par défaut de la directive de configuration PHP <code>register_globals</code> vaut off . La communauté PHP vous recommande de ne pas dépendre de cette directive, mais de trouver d'autres moyens pour accéder aux données, tels que les superglobals .

10.7.14 Comment puis-je générer des fichiers PDF sans utiliser les bibliothèques non libres

Il y'a quelques alternatives écrites en PHP tel que <http://www.ros.co.nz/pdf/>, <http://www.fpdf.org/>, <http://www.gnuvox.com/pdf4php/>, et <http://www.potentialtech.com/ppl.php>. Il existe aussi le module [Panda](#).

10.7.15 J'essaye d'accéder à une des variables standard CGI (comme

Il est important de réaliser que la directive PHP `register_globals` affecte aussi les variables d'environnement et de serveur. Lorsque `register_globals = off` (valeur par défaut depuis PHP 4.2.0), `$DOCUMENT_ROOT` n'existera pas. A la place, utilisez `$_SERVER['DOCUMENT_ROOT']`. Si `register_globals = on` alors les variables `$DOCUMENT_ROOT` et `$GLOBALS['DOCUMENT_ROOT']` existeront aussi.

Si vous êtes sûrs que `register_globals = on` et que vous vous demandez pourquoi `$DOCUMENT_ROOT` n'est pas disponible à l'intérieur de votre fonction, c'est parce que elle est comme toute autre variable et requiert donc global `$DOCUMENT_ROOT` dans le corps de la fonction. Voyez aussi la page du manuel à propos de la [portée des variables](#). Il est recommandé de coder avec `register_globals = off`.

Note
Superglobales : disponibilité

Depuis PHP 4.1.0, les tableaux superglobaux tels que `$_GET` , `$_POST` et `$_SERVER` , etc. sont disponibles. Pour plus d'informations, lisez la section [superglobals](#)

10.7.16 Quelques directives PHP peuvent également prendre des valeurs d'octets sténographiées contrairement aux valeurs d'octets uniquement

Les options disponibles sont K (pour Kilo octets) et M (pour Méga octets) et G (pour Giga octets ; disponible depuis PHP 5.1.0), ils sont insensibles à la casse. Tout autre chose est supposé représenter des octets. 1M équivaut à un Méga octets ou 1048576 octets. 1K équivaut à un Kilo octets ou 1024 octets. Vous ne devriez pas utiliser ces notations sténographiques en dehors du `php.ini` , à la place, utilisez une valeur entière en octets. Lisez la documentation sur la fonction `ini_get` pour un exemple sur la façon de convertir ces valeurs.

10.8 PHP et HTML

PHP et HTML sont très interactifs : PHP peut générer du HTML et HTML peut passer des informations à PHP. Avant de lire cette FAQ (foire aux questions), il est important que vous appreniez comme [récupérer des variables externes à PHP](#) . La page du manuel correspondante contient beaucoup d'exemples. Faites particulièrement attention à ce que signifie `register_globals` .

10.8.1 Quel encodage/décodage ai-je besoin lors du passage d'une valeur via un formulaire/une URL ?

Il y a plusieurs étapes pour lesquelles le codage est important. En supposant que vous avez une chaîne de caractères `$data` , qui contient la chaîne que vous voulez passer de manière non-encodée, voici les étapes appropriées :

- Interprétation HTML. Afin d'indiquer une chaîne aléatoire, vous **devez** l'inclure entre doubles guillemets et utiliser la fonction `htmlspecialchars` pour encoder la chaîne.
- URL : une URL est constituée de plusieurs parties. Si vous voulez que vos données soient interprétées comme un seul élément, vous **devez** les encoder avec la fonction `urlencode` .

Un élément de formulaire HTML caché

```
<?php
echo '<input type="hidden" value="' . htmlspecialchars($data) . '" />';
?>
```

Note

Il n'est pas correct d'utiliser la fonction `urlencode` pour vos données `$data` , car il en est de la responsabilité du navigateur de les encoder. Tous les navigateurs populaires le font correctement. Notez que cela s'effectue sans considération de la méthode utilisée (i.e GET ou POST). Vous devez uniquement noter ce cas pour les requêtes GET , car les requêtes POST sont généralement cachées.

Données éditables par l'utilisateur

```
<?php
echo "<textarea name=\"mydata\">\n";
echo htmlspecialchars($data) . "\n";
echo '</textarea>';
?>
```

Note

Les données sont montrées dans le navigateur comme prévues, car celui-ci interprétera les symboles HTML échappés.

Au moment de la validation, via la méthode GET ou POST , les données devraient être url-encodées par le navigateur avant le transfert et directement url-décodées par PHP. Donc, finalement, vous n'avez pas à effectuer d'url-encodage/url-decodage vous-même, tout est effectué automatiquement.

Dans une URL

```
<?php
echo "<a href=\"\" . htmlspecialchars("/nextpage.php?stage=23&data=" .
    urlencode($data)) . "\">\n";
?>
```

Note

En fait, vous simulez une requête GET HTML, il est nécessaire d'utiliser manuellement la fonction urlencode sur vos données.

Note

Vous devez utiliser htmlspecialchars sur l'URL complète, car l'URL se comporte comme la valeur d'un attribut HTML. Dans ce cas, le navigateur fera un htmlspecialchars sur la valeur et passera le résultat à l'URL. PHP devrait comprendre l'URL correctement, car vous avez url-encodé les données.

Vous devez noter que & dans l'URL est remplacé par & . Bien que la plus part des navigateurs devraient corriger cela si vous l'oubliez, ce n'est pas toujours le cas. Donc, même si votre URL n'est pas dynamique, vous **devez** utiliser la fonction htmlspecialchars sur l'URL.

10.8.2 J'essaye d'utiliser <input type="image"> mais les variables

Lorsque vous validez un formulaire, il est possible d'utiliser une image au lieu du bouton standard de type " submit " avec une balise du type :

```
<input type="image" src="image.gif" name="foo" />
```

Lorsque l'utilisateur clique sur l'image, le formulaire est transmis au serveur avec deux variables supplémentaires : foo.x et foo.y qui représentent les coordonnées du point cliqué.

Comme foo.x et foo.y sont des noms de variables invalides en PHP, elles sont automatiquement converties en foo_x et foo_y . Les points sont remplacés par des soulignés . Donc, vous devez accéder à ces variables comme n'importe quelle autre variable tel que décrit dans la section " Variables externes à PHP ". Par exemple, en utilisant \$_GET['foo_x'] .

Note

Les espaces dans les noms de variables sont également converties en un souligné bas.

10.8.3 Comment créer un tableau dans une balise <form> HTML ?

Pour envoyer le résultat du <form> comme un tableau de variables à votre script PHP, vous devez nommer, via l'attribut name , les balises <input>, <select> ou <textarea> comme cela :

```
<input name="MonTableau[]" />
<input name="MonTableau[]" />
<input name="MonTableau[]" />
<input name="MonTableau[]" />
```

Noter les crochets après le nom de la variable, c'est ce qui fait que celle-ci sera un tableau. Vous pouvez grouper les éléments dans différents tableaux de variables en assignant le même nom à différents éléments :

```
<input name="MonTableau[]" />
<input name="MonTableau[]" />
<input name="MonAutreTableau[]" />
<input name="MonAutreTableau[]" />
```

Cela produira deux tableaux de variables, MonTableau et MonAutreTableau, qui seront envoyés au script PHP. Il est également possible d'assigner des clés spécifiques à votre tableau :

```
<input name="UnAutreTableau[]" />
<input name="UnAutreTableau[]" />
<input name="UnAutreTableau[email]" />
<input name="UnAutreTableau[telephone]" />
```

Le tableau UnAutreTableau contiendra les clés 0, 1, email et telephone.

Note
Le fait de spécifier une clé à un tableau est optionnel en HTML. Si vous ne le faites pas, les clés du tableaux suivront l'ordre d'apparition des éléments dans le formulaire. Dans notre premier exemple, le tableau contient les clés 0, 1, 2 et 3.

Voir aussi les [fonctions sur les tableaux de variables](#) et la section sur les [variables externes à PHP](#) .

10.8.4 Comment puis-je récupérer le résultat d'un champs HTML

Le champ SELECT multiple en HTML permet à l'utilisateur de sélectionner plusieurs éléments d'une liste. Ces éléments seront transmis à la page pointée par l'attribut action de la balise form . Le problème est que ces éléments sont tous passés avec le même nom de variable.

```
<select name="var" multiple="yes">
```

Chaque option sélectionnée arrivera au mécanisme de traitement sous la forme :

```
var=option1
var=option2
var=option3
```

Chaque option effacera donc le contenu de la précédente variable \$var . La solution consiste à utiliser un tableau de variables dans cet élément de formulaire HTML, par exemple :

```
<select name="var[]" multiple="yes">
```

Cela fera que PHP traitera \$var comme un tableau de variables et que chaque assignement de valeur à var[] ajoutera un index au tableau. La première option choisie sera mise dans \$var[0] , la suivante sera mise dans \$var[1] , etc. La fonction [count](#) peut être utilisée pour déterminer combien d'options ont été sélectionnées, et la fonction [sort](#) peut être utilisée pour trier le tableau, si nécessaire.

Notez que si vous utilisez Javascript, [] dans le nom de l'élément peut vous poser problème lorsque vous tenterez d'accéder à celui-ci par son nom. Utilisez plutôt l'indice numérique de l'élément dans ce cas, ou bien utilisez les simples guillemets pour entourer cet élément, comme :

```
variable = documents.forms[0].elements['var[]'];
```

10.8.5 Comment puis-je passer une variable de Javascript vers PHP ?

Javascript est (habituellement) une technologie coté client et PHP est (habituellement) une technologie coté serveur et sachant que HTTP est un protocole statique, les deux langages ne peuvent pas directement partager des variables.

Cependant, il est possible de faire passer des variables entre les deux. Une des solutions pour cela est de générer un code Javascript à l'aide de PHP et de faire rafraîchir le navigateur tout seul, passant ainsi des variables spécifiques au script PHP. L'exemple suivant montre précisément comme réaliser cela -- il permet à PHP de récupérer les dimensions de l'écran du client, ce qui est normalement possible qu'en technologie coté client.

```
<?php
if (isset($_GET['largeur']) AND isset($_GET['hauteur'])) {
    // Affichage des variables
    echo 'La largeur de l\'écran est : ' . $_GET['largeur'] . "<br />\n";
    echo 'La hauteur de l\'écran est : ' . $_GET['hauteur'] . "<br />\n";
} else {
    // passage des variables de dimensions
    // (préservation de la requête d'origine
    // -- les variables par méthode POST doivent être traitées différemment)

    echo "<script type=\"text/javascript\">\n";
    echo "    location.href=\"${_SERVER['SCRIPT_NAME']}?${_SERVER['QUERY_STRING']}\"
        . "&largeur=\" + screen.width + \"&hauteur=\" + screen.height;\n";
    echo "</script>\n";
    exit();
}
?>
```

10.9 PHP et COM

PHP peut être utilisé pour accéder à des objets COM et DCOM sur les plates-formes Win32.

10.9.1 J'ai compilé une DLL pour calculer quelque chose. Y a-t-il un moyen d'utiliser cette DLL sous PHP?

Si c'est une DLL simple, il n'y a aucun moyen pour le moment de l'utiliser avec PHP. Si la DLL contient un serveur COM, vous pourrez l'utiliser si elle implémente l'interface IDispatch.

10.9.2 Que signifie 'Unsupported variant type: xxxx (0xxxxx)'?

Il y a des dizaines de types de VARIANT et de combinaisons entre elles. La plupart d'entre elles sont déjà supportées, mais quelques une ne sont toujours pas implémentées. Les tableaux ne sont pas complètement supportés. Seuls le tableaux unidimensionnels indexés peuvent être transmis entre PHP et COM. Si vous trouvez d'autres types qui ne sont pas supportés, reportez les nous comme un bogue (si ce n'est pas déjà fait) et fournissez le plus d'informations possibles.

10.9.3 Est-il possible de manipuler des objets visuels en PHP?

Généralement, c'est possible, mais comme PHP est utilisé le plus souvent en tant que langage de script web et dans un environnement de serveur web, les objets visuels n'apparaîtront jamais sur le bureau du serveur. Si vous voulez utiliser PHP pour scripter des applications, par exemple avec PHP-GTK, il n'y a aucune limitation à accéder et manipuler des objets visuels via COM.

10.9.4 Puis-je stocker un objet COM dans une session?

Non, vous ne pouvez pas. Les instances COM sont traitées comme des ressources, ce qui signifie qu'elles ne sont disponibles que dans un seul contexte de script.

10.9.5 Comment puis-je intercepter des erreurs COM?

En PHP 5, l'extension COM envoie des exceptions `com_exception`, que vous pouvez intercepter en inspectant le membre `code` pour déterminer que faire.

En PHP 4, il n'est pas possible d'intercepter des erreurs COM en dehors de la façon dont elles sont fournies par PHP lui-même (`@`, `track_errors`, ..).

10.9.6 Puis-je générer des fichiers DLL à partir de PHP comme je le fais avec Perl?

Non, il n'y a malheureusement pas d'outil disponible pour le faire en PHP.

10.9.7 Que signifie 'Unable to obtain IDispatch interface for CLSID {xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}'?

Cette erreur peut avoir plusieurs causes :

- le CLSID est incorrect
- la DLL demandée est introuvable
- le composant demandé n'implémente pas l'interface IDispatch

10.9.8 Comment puis-je utiliser des objets COM à partir d'un serveur distant?

Exactement de la même manière qu'avec des objets locaux. Vous devez juste passer l'adresse IP de la machine distante en deuxième paramètre du constructeur COM.

Assurez-vous que `com.allow_dcom = TRUE` dans votre `php.ini`.

10.9.9 J'obtiens 'DCOM is disabled in C:\path...\scriptname.php on line 6', que dois-je faire?

Editez votre php.ini et mettez `com.allow_dcom = TRUE` .

10.9.10 Est-il possible de charger/manipuler des objets ActiveX dans une page en PHP ?

Cela n'a rien à voir avec PHP. Les objets ActiveX sont chargés côté client s'il sont demandés par le document HTML. Il n'y a aucune relation avec le script PHP et il n'y a pas d'interaction directe possible avec le serveur.

10.9.11 Est-il possible d'obtenir une instance fonctionnelle d'un composant?

C'est possible avec l'aide de monikers. Si vous voulez des références multiples aux même mot d'instance, vous pouvez créer une instance de la façon suivante :

```
<?php
$word = new COM("C:\docs\word.doc");
?>
```

Cela créera une nouvelle instance s'il n'y en a aucune disponible ou retournera un descripteur vers l'instance courante, si elle est disponible.

10.9.12 Y a-t-il moyen de manipuler un événement envoyé par un objet COM?

Vous pouvez définir un moniteur d'événement (sink) et le lier en utilisant `com_event_sink` . Vous pouvez utiliser `com_print_typeinfo` pour que PHP génère un squelette pour la classe du moniteur d'événement.

10.9.13 J'ai des problèmes quand j'invoque une méthode d'un objet COM qui expose plus d'une interface. Que puis-je faire?

La réponse est aussi simple que non satisfaisante. Je ne sais pas exactement, mais je pense que vous ne pouvez rien faire. Si quelqu'un a des informations spécifiques sur ce sujet, faites le moi savoir :)

10.9.14 Bon, PHP fonctionne avec COM, mais qu'en est-il de COM+?

COM+ étend COM avec un canevas pour gérer des composants via MTS et MSMQ mais il n'y a rien de particulier que PHP doit supporter pour utiliser de tels composants.

10.9.15 Si PHP peut manipuler des objets COM, peut-on imaginer d'utiliser des ressources de composants, en conjonction avec PHP?

PHP ne supporte pas encore les transactions. Ainsi, si une erreur se produit, aucun rollback n'est initié. Si vous utilisez des composants qui supportent les transactions, vous devrez implémenter le gestionnaire de transactions par vous-même.

10.10 PHP et les autres langages

PHP est le meilleur langage pour la programmation web, mais qu'en est-il des autres langages?

10.10.1 PHP contre ASP?

ASP n'est pas vraiment un langage en lui-même, c'est l'acronyme de Active Server Pages, le langage courant pour programmer en ASP étant le Visual Basic ou le JScript. Le plus gros inconvénient de ASP est que c'est un système propriétaire utilisé nativement sur Microsoft Internet Information Server (IIS). Cela limite sa disponibilité aux seules plates-formes Win32. Il existe cependant des projets pour utiliser ASP dans d'autres environnements et avec d'autres serveurs web : [InstantASP](#) de [Halcyon](#) (commercial), Chili!Soft ASP de [Chili!Soft](#) (commercial). ASP est connu pour être un langage plus lent et plus lourd que PHP, mais aussi moins stable. Les gens favorables à ASP prétendent que comme il utilise de base VBScript, c'est un langage facile à apprendre si vous connaissez déjà Visual Basic. Le support de ASP est activé par défaut dans IIS, e rendant facile à mettre en oeuvre et à utiliser. Les composants inclus de base dans ASP sont vraiment limités, et si vous devez utiliser des fonctionnalités "avancées" comme l'interaction avec des serveurs FTP, vous devrez acheter des composants supplémentaires.

10.10.2 Existe-t-il un convertisseur de ASP vers PHP?

Oui, le programme côté serveur [asp2php](#) est le plus fréquemment utilisé, de même que cette option [côté client](#) .

10.10.3 PHP contre Cold Fusion?

PHP est réputé pour être plus rapide et plus efficace pour la programmation complexe et pour tester de nouvelles idées. PHP est aussi plus stable et consomme moins de ressources. Cold Fusion a une meilleure gestion des erreurs, une meilleure abstraction de bases de données et une meilleure gestion des dates, bien que l'abstraction de bases de données soit chose possible en PHP 4. Un autre point réputé pour être une des forces de Cold Fusion est sont excellent moteur de recherche, mais il peut paraitre absurde d'intégrer un moteur de recherche dans un langage de script Web. PHP fonctionne sur nombre de plates-formes alors que Cold Fusion ne supporte que Win32, Solaris, Linux et HP/UX. Cold Fusion intègre un bon environnement de développement (IDE) et il est facile à apprendre, alors que PHP requiert davantage de connaissances en programmation. Cold Fusion est pensé pour des non-développeurs, alors que PHP est pensé pour les développeurs.

Un excellent résumé de Michael J Sheldon sur ce sujet a été posté sur la liste de diffusion PHP. Une copie du message peut être trouvé sur <http://marc.theaimsgroup.com/?l=php-general&m=95602167412542&w=1> .

10.10.4 PHP contre Perl?

Le plus grand avantage de PHP par rapport à Perl est que PHP a été pensé pour la programmation Web alors que Perl a été pensé pour faire des choses bien plus diversifiées, ce qui le rend bien plus compliqué. La complexité / flexibilité de Perl rend l'écriture de script plus facile, mais en rend la lecture par un autre auteur / codeur beaucoup moins aisée. PHP est plus facile que Perl à intégrer dans du HTML. PHP a toutes les "bonnes" fonctionnalités de Perl : la construction, la syntaxe... sans pour autant être aussi compliqué que Perl. Perl est un véritable langage bien testé, du fait de son existence depuis la fin des années 80, mais PHP devient mature très rapidement.

10.11 Migrer de PHP 2 à PHP 3

PHP a déjà une longue histoire derrière lui : Le légendaire PHP 1.0, PHP/FI, PHP 3.0 et PHP 4.0.

10.11.1 Migrer de PHP 2 à PHP 3 ?

PHP/FI 2.0 n'est plus supporté. Reportez-vous à [la section appropriée du manuel](#) pour des informations à propos de la migration à partir de PHP/FI 2.0.

Si vous travaillez encore avec PHP 2, nous vous recommandons **vivement** de mettre à jour en PHP 4.

10.12 Migrer de PHP 3 à PHP 4

PHP a déjà une longue histoire derrière lui : Le légendaire PHP 1.0, PHP/FI, PHP 3.0 et PHP 4.0.

10.12.1 Migrer de PHP 3 à PHP 4

PHP 4 a été conçu pour être aussi compatible avec les anciennes versions de PHP que possible et très peu de fonctionnalités ont changées dans le processus. Si vous n'êtes pas sûrs de la compatibilité, vous devez installer PHP 4 dans un environnement de test et y tester vos scripts.

Lisez aussi [l'appendice de la migration approprié](#) de ce manuel.

10.12.2 Les sessions fonctionnent-elles avec PHP 3 ?

Même si le [support natif des sessions](#) n'existait pas en PHP 3, il y'a des applications tierces qui ont (et encore aujourd'hui) offert les fonctionnalités des sessions. La méthode la plus commune est d'utiliser [PHPLIB](#) .

10.12.3 Fonctions incompatibles ?

Puisque PHP 4 est basiquement une réécriture entière du moteur PHP il y a peu de fonctions qui ont été altérées et celles qui l'ont été faisaient partie des fonctions les plus exotiques.

10.13 Migration de PHP 4 à PHP 5

Cette section devrait vous aider à migrer de PHP 4 à PHP 5.

10.13.1 Migration de PHP 4 à PHP 5

Bien que PHP 5 offre beaucoup de nouvelles fonctionnalités, il a été prévu pour être compatible avec les anciennes versions de PHP autant que possible, avec un minimum de fonctionnalités rompues dans le processus.

Assurez-vous de lire l'annexe intitulé "[Migrer en PHP 5](#)" de ce manuel ; il contient beaucoup plus d'informations concernant la migration en PHP 5.

10.13.2 Est-ce que MySQL fonctionne en PHP 5 ? Il semble avoir disparu.

[MySQL](#) est supporté avec la seule différence que le support de MySQL n'est plus activé par **défaut** en PHP 5. Cela signifie essentiellement que PHP n'inclue pas automatiquement `--with-mysql` dans la [configuration](#) ; vous devez donc maintenant l'ajouter à la main lorsque vous compilez PHP. Les utilisateurs de Windows doivent éditer leur `php.ini` et activer la bibliothèque DLL `php_mysql.dll` , sachant qu'en PHP 4, cette bibliothèque n'existait pas, elle était automatiquement incluse dans le binaire PHP.

En outre, la bibliothèque cliente MySQL n'est plus incluse avec PHP. Plus de détails sur ce sujet sont consultables dans cette [section de la FAQ](#) ; lisez également la [section MySQL](#) pour les détails concernant l'installation de MySQL. Un exemple de ligne de configuration serait `--with-mysql=/usr` alors que les utilisateurs de Windows doivent avoir la bibliothèque DLL nommée `libmysql.dll` de disponible sur leur système.

10.13.3 J'ai entendu dire que PHP 5 à un tout nouveau model objet ; est-ce que mon actuel code objet fonctionne ? Où puis-je trouver des informations concernant ces nouvelles fonctionnalités ?

Le principal changement en PHP 5 est le nouveau model objet car PHP 5 utilise désormais le **Zend Engine 2.0** . La directive `zend.ze1_compatibility_mode` active la compatibilité avec le **Zend Engine 1.0** (PHP 4).

Le nouveau model objet est documenté dans la [référence du langage orienté objet](#) ainsi que dans les sections concernant la [migration du model objet](#) .

10.13.4 Donc, mise à part le nouveau model objet, qu'est-ce qui a changé dans PHP 5 ? En particulier, existe-t-il une version spécifique du manuel PHP ?

Peu de changements existent mise à part le model objet, lisez l'annexe "[Migration 5](#)" pour plus de détails. Il n'y a pas de version spécifique à PHP 5 de ce manuel car la majorité des fonctionnalités de PHP restent les mêmes.

10.14 Questions diverses

Il y'a quelques questions inclassables. Vous les trouverez ici.

10.14.1 Comment puis-je manipuler le manuel comprimé en bzip2 sous Windows ?

Si vous n'avez pas d'outil d'archivage prenant en charge les fichiers au format bz2, [téléchargez](#) l'outil en ligne de commande à partir de Redhat (merci de lire les informations plus bas).

Si vous préférez ne pas utiliser d'outil en ligne de commande, vous pouvez essayer des outils gratuits comme [Stuffit Expander](#) , [UltimateZip](#) , [7-Zip](#) , ou [Quick Zip](#) . Si vous avez des outils comme [WinRAR](#) ou [Power Archiver](#) , vous pouvez facilement décompresser les fichiers bz2 avec. Si vous utilisez Total Commander (autrefois Windows Commander), un plugin bz2 est disponible gratuitement sur le site de [Total Commander](#) .

L'outil en ligne commande bzip2 de Redhat :

Les utilisateurs du service pack 2 de Win2K doivent prendre la version 1.0.2, tous les autres utilisateurs de Windows doivent prendre la version 1.00. Après avoir téléchargé, renommez l'exécutable en bzip2.exe. Pour votre convenance, mettez-le dans un dossier dans votre path, c'est à dire C:\Windows où C représente votre disque où Windows est installé.

Note : Dans ce qui suit, lang représente votre langue et x le format désiré, par exemple : pdf. Pour décompresser php_manual_lang.x.bz2 suivez les instructions suivantes :

- ouvrez un invite de commande
- déplacez-vous dans le dossier où se trouvent les fichiers php_manual_lang.x.bz2
- lancez la commande `bzip2 -d php_manual_lang.x.bz2`, pour extraire php_manual_lang.x dans ce même dossier

Dans le cas où vous avez téléchargé le fichier php_manual_lang.tar.bz2 avec beaucoup de fichiers HTML à l'intérieur, la procédure est la même. La seule différence est que vous obtiendrez un fichier php_manual_lang.tar. Le format tar est connu pour être pris en compte par la plupart des archiveurs de Windows, comme [WinZip](#) .

10.14.2 Que signifie le caractère "&" devant les arguments dans les déclarations des fonctions comme la fonction

Cela signifie que cet argument est passé par référence et que la fonction le modifiera, comme expliqué dans la documentation. Vous pouvez passer uniquement des variables de cette façon et vous n'avez pas besoin de les passer avec le caractère "&" dans l'appel de la fonction (car déprécié).

10.14.3 Comment puis-je gérer la directive

Pour plus d'informations concernant l'implication sur la sécurité de la directive `register_globals` , lisez le chapitre sur la sécurité concernant l' utilisation de register_globals .

Il est préférable d'utiliser les superglobals , plutôt que de remettre à on la directive `register_globals` .

Si vous êtes sur un serveur mutualisé avec la directive `register_globals` de positionnée à `off` et que vous devez utiliser des applications qui requièrent que cette option soit à `on` ou si vous êtes hébergés sur un serveur qui a cette directive d'activée mais que vous voudriez éliminer ce risque sécuritaire, vous devriez émuler l'argument opposé avec PHP. C'est toujours une bonne idée que de demander tout d'abord s'il est possible de modifier cette option dans la configuration de PHP mais si cela n'est pas possible, vous pouvez utiliser ces astuces de compatibilité.

Emulation des Register Globals

Ceci émuler la directive `register_globals` à `On` .

```
<?php
// Emulation de register_globals à on
if (!ini_get('register_globals')) {
    $superglobals = array($_SERVER, $_ENV,
        $_FILES, $_COOKIE, $_POST, $_GET);
    if (isset($_SESSION)) {
        array_unshift($superglobals, $_SESSION);
    }
    foreach ($superglobals as $superglobal) {
        extract($superglobal, EXTR_SKIP);
    }
}
?>
```

Ceci émuler `register_globals` à `Off`. Gardez à l'esprit que ce code doit être appelé au tout début de votre script.

```
<?php
// Emulation de register_globals à off
function unregister_GLOBALS()
{
    if (!ini_get('register_globals')) {
        return;
    }

    // Vous pouvez vouloir modifier cela pour avoir une erreur plus jolie
    if (isset($_REQUEST['GLOBALS']) || isset($_FILES['GLOBALS'])) {
        die('Tentative d\'effacement des GLOBALS détectée');
    }

    // Les variables à ne jamais effacer
    $noUnset = array('GLOBALS', '_GET',
        '_POST', '_COOKIE',
        '_REQUEST', '_SERVER',
        '_ENV', '_FILES');

    $input = array_merge($_GET, $_POST,
        $_COOKIE, $_SERVER,
        $_ENV, $_FILES,
        isset($_SESSION) && is_array($_SESSION) ? $_SESSION : array());

    foreach ($input as $k => $v) {
        if (!in_array($k, $noUnset) && isset($GLOBALS[$k])) {
            unset($GLOBALS[$k]);
        }
    }
}

unregister_GLOBALS();
?>
```

11 Annexes

11.1 Histoire de PHP

L'évolution de PHP s'est faite en quelques années. Devenir l'un des langages les plus importants du web ne fut pas une évolution simple. Pour ceux que ça intéresse, voici comment PHP a évolué jusqu'à aujourd'hui. Les vieilles versions de PHP sont disponibles dans le [musée PHP](#) .

11.1.1 Histoire de PHP

11.1.1.1 PHP/FI

PHP a pris la suite d'un langage plus ancien, appelé PHP/FI. PHP/FI a été créé par Rasmus Lerdorf, en 1995. C'était initialement une bibliothèque de scripts Perl, dont il se servait pour noter les accès à son CV en ligne. Il donna le nom de 'Personal Home Page Tools' à cette bibliothèque. Au fur et à mesure qu'il ajoutait de nouvelles fonctionnalités, Rasmus a transformé la bibliothèque en une implémentation en C, capable de communiquer avec les bases de données, et de créer des applications dynamiques et simples pour le web. Rasmus décida alors de publier [son code](#) , pour que tout le monde puisse l'utiliser et en profiter. Cela appela aussi aux contributions et aux améliorations du code.

PHP/FI, qui signifie Personal Home Page / Forms Interpreter, (Home Page personnelle, Interpréteur de Formulaires), incluait plusieurs fonctionnalités de base que nous connaissons encore aujourd'hui. Il avait ces variables qui ressemblent au Perl, un système d'interprétation automatique des variables de formulaires, et une syntaxe qui s'intègre facilement dans HTML. La syntaxe elle-même était similaire à celle du Perl mais beaucoup plus limitée. Elle était simple et un peu incohérente.

En 1997, PHP/FI 2.0, la seconde version en langage C, avait déjà une audience estimée de plusieurs milliers d'utilisateurs dans le monde, et environ 50,000 noms de domaine indiquaient qu'ils avaient installé PHP. Cela représentait environ 1% des noms de domaine sur l'Internet. Même si le nombre de contributeurs était plutôt élevé, PHP était toujours le projet d'un seul homme.

PHP/FI 2.0 fut publié officiellement en novembre 1997, après avoir passé l'essentiel de sa vie en version bêta. Peu de temps après, une version alpha de PHP 3.0 était publiée.

11.1.1.2 PHP 3

PHP 3.0 fut la première version du langage tel que nous le connaissons actuellement. Il fut créé par Andi Gutmans et Zeev Suraski en 1997, sous forme de réécriture complète de PHP/FI, lorsqu'ils s'aperçurent que PHP/FI était sous-performant pour leur application de commerce électronique. Dans un effort de coopération et de compatibilité avec les anciennes versions de PHP/FI, Andi, Rasmus et Zeev décidèrent de coopérer et d'annoncer PHP 3.0 comme le successeur officiel de PHP/FI. Le développement de PHP/FI 2.0 fut complètement arrêté.

Une des améliorations notables de PHP 3.0 fut ses capacités d'extensions. En plus de fournir une solide infrastructure aux utilisateurs finaux, des accès à de nombreuses bases de données et protocoles, PHP 3.0 proposait une API modulaire, qui attira des douzaines de développeurs. Ceux-ci réalisèrent et partagèrent de nouvelles extensions. Sans doute, ce fut la clé du succès

retentissant de PHP 3.0. Les autres améliorations de PHP 3.0 furent le support de la syntaxe objet, et une syntaxe de langage plus robuste et cohérente.

Le nouveau langage fut publié sous un nouveau nom, qui indiquait clairement que le projet n'était plus un projet personnel, comme l'était PHP/FI 2.0. Il fut nommé 'PHP' avec une nouvelle signification : 'PHP: Hypertext Preprocessor'. C'est un acronyme récursif, c'est-à-dire qu'il se définit lui-même. En français, cela donne : 'Le préprocesseur Hypertexte, c'est PHP'.

A la fin de 1998, PHP avait conquis une base de plusieurs dizaines de milliers d'utilisateurs, et des centaines de milliers de sites indiquaient qu'ils l'utilisaient. Au plus fort de son utilisation, PHP 3.0 était installé sur 10% du parc mondial de serveurs web.

PHP 3.0 fut officiellement publié en Juin 1998, après 9 mois de tests.

11.1.1.3 PHP 4

Durant l'hiver 1998, juste après la publication de PHP 3.0, Andi Gutmans et Zeev Suraski commencèrent la réécriture du moteur interne de PHP à la base. L'objectif était d'améliorer les performances de PHP avec les applications complexes et d'améliorer la modularité du code. Ces applications étaient rendues possibles par la syntaxe de PHP 3.0 mais le logiciel n'était pas conçu pour supporter efficacement ces applications.

Le nouveau moteur, appelé 'Zend Engine' (combinaison des noms de Zeev et Andi), atteint ces objectifs avec succès, et la première version fut publiée vers la mi-1999. PHP 4.0, s'appuyant sur ce moteur et amélioré par un grand nombre de nouvelles fonctionnalités, fut publié officiellement en mai 2000, presque 2 ans après son prédécesseur. En plus de performances nettement plus élevées, PHP 4.0 apportait le support de nombreux serveurs web, les sessions HTTP, la bufferisation de sortie, une sécurité accrue des informations visiteurs et plusieurs nouvelles structures de langage.

Actuellement, PHP est utilisé par des centaines de milliers de développeurs, et plusieurs millions de sites web indiquent qu'ils sont configurés avec PHP, ce qui représente environ 20% des noms de domaine sur Internet.

L'équipe de développement de PHP inclut des douzaines de développeurs, et d'autres équipes travaillent à des projets liés tels que PEAR ou la documentation.

11.1.1.4 PHP 5

PHP 5 est sorti en Juillet 2004 après un long développement et plusieurs pré-versions. Il est régi par son moteur, le Zend Engine 2.0 avec un nouveau modèle et des dizaines de nouvelles fonctionnalités. Pour plus d'informations concernant ce moteur, lisez [cette page web](#) .

11.1.2 Quelques projets liés à PHP

11.1.2.1 PEAR

PEAR , signifie ' PHP Extension and Application Repository ' (initialement, PHP Extension and Add-on Repository) et est la première version des classes de base en PHP. Elle deviendra à terme le moyen privilégié de distribuer des extensions PHP entre développeurs.

PEAR a été conçu durant les débats tenus lors des PHP Developers' Meeting (PDM) (Rencontres de développeurs PHP), qui ont eu lieu en Janvier 2000 à Tel Aviv. Il a été créé à l'initiative de Stig S. Bakken, et dédié à sa première fille, Malin Bakken.

Depuis l'an 2000, PEAR a grossi jusqu'à devenir un projet significatif avec un groupe de développeurs compétents, qui travaillent à mettre en place une bibliothèque complète, réutilisable et commune à la communauté PHP entière. PEAR inclut une vaste sélection de classes de base pour réaliser une couche d'abstraction de bases de données, du cache, des calculs mathématiques et des transactions de commerce électronique.

Plus d'informations sur PEAR sont disponibles dans [le manuel](#) .

11.1.2.2 Equipe d'assurance Qualité

L'équipe d'assurance Qualité de PHP a été mise en place durant l'été 2000, en réponse aux critiques reçues par PHP 3.0, qui n'était pas suffisamment testé sur des environnements de production. L'équipe est constituée d'un groupe de développeurs de haut niveau, qui ont une bonne connaissance des sources PHP. Ces développeurs passent le plus clair de leur temps à localiser et supprimer les bogues trouvés dans PHP. De plus, il y a bien d'autres membres du groupe PHP qui testent et fournissent un compte-rendu fidèle de l'utilisation de PHP sur différentes plate-formes.

11.1.2.3 PHP-GTK

PHP-GTK est la solution PHP pour écrire des applications avec interface, coté client. Andrei Zmievski se souvient de la création de PHP-GTK :

La programmation d'interface GUI a toujours été une passion pour moi, et je pensais que Gtk+ était une excellente bibliothèque, hormis le fait qu'elle était écrite en C, ce qui était plutôt laborieux. Après avoir assisté aux implémentations de PyGtk et GTK-Perl, j'ai décidé de voir si PHP pouvait disposer de sa propre interface avec Gtk+, même minimale. J'ai commencé au mois d'août 2000, alors que j'avais un peu plus de temps libre, et j'ai réalisé les premières expérimentations. Mon guide principal fut l'implémentation de PyGtk, qui était plutôt complète et avait une interface orientée objet. James Henstridge, l'auteur de PyGtk, fut d'une aide précieuse au démarrage du projet.

Réécrire toutes les interfaces de Gtk+ à la main était hors de question et j'ai pensé à réaliser un générateur de code, exactement comme PyGtk l'avait fait. Le générateur de code était un programme PHP qui lisait un ensemble de fichiers .defs , contenant les classes, constantes et méthodes Gtk+, puis générait le code d'interface de PHP. Ce qui ne pouvait pas être généré automatiquement était traité à la main, dans le fichier .overrides .

Travailler sur le générateur de code et sur l'infrastructure prit un peu de temps, car je n'avais pas beaucoup de temps à consacrer à PHP-GTK durant l'automne 2000. Après que j'ai montré PHP-GTK à Frank Kromann, il s'y intéressa et commença à m'aider sur le générateur et l'implémentation Win32. Lorsque nous écrivîmes le premier programme 'Hello World', ce fut extrêmement excitant. Cela a pris encore quelques mois pour avoir une version initiale. Elle fut publiée en mars 2001. Cela a encore pris quelques mois pour que le projet soit dans une forme présentable. Ce projet fut rapidement présenté sur SlashDot.

Anticipant que PHP-GTK serait un projet à part entière, j'ai installé des listes de diffusion et un serveur CVS indépendants, de même que le site gtk.php.net, avec l'aide de Colin Viebrock. La documentation demandait aussi de l'aide, et James Moore vint me porter une aide précieuse.

Depuis sa publication PHP-GTK a gagné en popularité. Nous avons notre propre équipe de documentation, et le manuel s'améliore de plus en plus. Des extensions ont été écrites pour PHP-GTK et des applications de plus en plus importantes sont conçues.

11.1.3 Livres traitant de PHP

Comme PHP a évolué, il a été reconnu comme une plate-forme de développement populaire. Un des signes qui ne trompent pas est que le nombre de livres dédiés à PHP a évolué parallèlement.

A notre connaissance, le premier livre dédié à PHP fut 'PHP - tvorba interaktivních internetových aplikací' - un livre tchèque publié en avril 1999, écrit par Jirka Kosek. Le mois suivant parut un livre allemand écrit par Egon Schmid, Christian Cartus et Richard Blume. Le premier livre en anglais sur PHP fut publié juste après : 'Core PHP Programming' par Leon Atkinson. Ces trois livres se rapportaient à PHP 3.0.

Même si ces livres restent uniques dans leur genre, ils furent rapidement suivis par un grand nombre d'autres livres, de différents éditeurs. Il y a plus de 40 livres en anglais, 50 en allemand et plus de 20 en français. De plus, on peut maintenant trouver des livres sur PHP dans de nombreuses autres langues, comme l'espagnol, le coréen, le japonais ou l'hébreu.

Ce grand nombre de livres, écrits par différents auteurs et publiés par différents éditeurs, ainsi que leur disponibilité en différentes langues, témoigne clairement du succès planétaire de PHP.

11.1.4 Publications sur PHP

A notre connaissance, le premier article consacré à PHP dans un magazine papier fut publié dans un magazine français, vers la fin 1998 et se rapportait à PHP 3.0. Comme pour les livres, ce fut le premier d'une longue série d'articles publiés dans différents magazines.

Des articles sur PHP ont été publiés dans Dr. Dobbs, Linux Enterprise, Linux Magazine et bien d'autres. Il existe même des articles sur la migration d'applications ASP vers PHP sous Windows dans la bibliothèque MSDN de Microsoft .

11.2 Migration de PHP 4 vers PHP 5

11.2.1 Nouveautés de PHP 5

Les modifications apportées à PHP 5 et au Zend Engine 2 augmentent considérablement les capacités et les performances de PHP. Une attention toute particulière a été apportée à ce que cette nouvelle version soit la plus compatible possible avec les scripts antérieurs. Ainsi la migration de votre code de PHP 4 vers PHP 5 devrait être aisée. La plupart des scripts PHP 4 devraient être prêts à fonctionner sans nécessiter la moindre modification. Il existe toutefois quelques différences et vous devriez tester vos codes avant de changer de version en production.

11.2.2 Incompatibilités avec les version antérieures

Bien que la plupart des scripts PHP 4 existants devraient fonctionner, il convient de noter quelques incompatibilités avec les versions antérieures de PHP :

- Il y a quelques nouveaux mots réservés .
- strpos et stripos utilisent maintenant une chaîne de caractères complète (un seul caractère auparavant) en tant qu'élément de recherche.
- L'utilisation d'index illégaux sur une chaîne de caractères entraîne maintenant un message E_ERROR au lieu d'un message E_WARNING auparavant. Voici un exemple incorrect : `$str = 'abc'; unset($str[0]);` .
- La fonction array_merge a été modifiée pour n'accepter que des tableaux. Pour chaque variable passée en paramètre autre qu'un tableau, un message E_WARNING sera envoyé. Soyez attentifs car votre script pourrait émettre des E_WARNING de façon inopinée.
- La variable de serveur PATH_TRANSLATED n'est plus définie implicitement avec Apache2 SAPI contrairement à auparavant (PHP 4) où elle était fixée avec la même valeur que SCRIPT_FILENAME lorsqu'Apache ne la renseignait pas. Cette modification a été apportée afin d'être en conformité avec les spécifications CGI . Merci de consulter le bogue #23610 pour plus d'informations, ainsi que la description de \$_SERVER['PATH_TRANSLATED'] du manuel. Ce problème affecte également PHP >= 4.3.2.
- La constante T_ML_CONSTANT n'est plus définie par l'extension Tokenizer . Si error_reporting est réglé à E_ALL , PHP va générer un avertissement. Bien que T_ML_CONSTANT n'ait jamais été utilisée, elle était définie dans PHP 4. Avec PHP 4 et PHP 5, // et /* */ sont assimilés à la constante T_COMMENT . Toutefois les commentaires de style PHPDoc /** */ , qui sont analysés depuis PHP 5, sont reconnus en tant que T_DOC_COMMENT .
- \$_SERVER contient dorénavant argc et argv si votre variables_order inclut "S". Si vous avez configuré votre système pour qu'il ne crée pas \$_SERVER , ils n'existeront bien sûr pas. Cette modification a été effectuée afin que argc et argv soient toujours accessibles dans la version CLI quelle que soit la valeur de variables_order . Ainsi, la version CLI renseignera dorénavant toujours les variables \$argc et \$argv .
- Un objet sans propriété n'est plus considéré comme vide (empty).
- Dans certains cas, les classes doivent être déclarées avant d'être utilisées. Cela survient uniquement si les nouvelles fonctionnalités de PHP 5 sont utilisées. Sinon, le comportement sera le même qu'avant.
- Les fonctions get_class , get_parent_class et get_class_methods retournent désormais le nom de la classe comme elle a été déclarée (sensible à la casse), ce qui peut causer des problèmes dans vos anciens scripts qui utilisent le comportement précédent (le nom de la classe était toujours retourné en minuscules). Une solution possible est de rechercher ces fonctions dans tous vos anciens scripts et d'utiliser la fonction strtolower .

Ces changements de sensibilité à la casse sont également appliqués aux constantes magiques prédéfinies __CLASS__ , __METHOD__ et __FUNCTION__ . Les valeurs sont retournées exactement comme elles ont été déclarées (sensible à la casse).

- La fonction ip2long retourne maintenant FALSE lorsqu'une adresse IP invalide est entrée comme argument de la fonction, et non plus -1 .
- Lorsque des fonctions sont définies dans un fichier inclus, elles peuvent être utilisées dans le fichier principal, qu'elles aient été définies avant ou après l'instruction return . Si le fichier est inclus deux fois, PHP 5 émet une erreur fatale car les fonctions sont toujours déclarées, tandis que PHP 4 n'a aucun problème avec ça. Il est recommandé d'utiliser l'instruction include_once au lieu de vérifier si le fichier a déjà été inclus et de retourner, conditionnellement, l'inclusion du fichier.
- include_once et require_once normalise d'abord le chemin du fichier à inclure sous Windows, donc inclure le fichier A.php et le fichier a.php revient à n'inclure qu'une seule fois le fichier.

strpos et stripos recherchent maintenant une chaîne de caractères dans une autre

```
<?php
var_dump(strpos('ABCDEF', 'DEF')); //int(3)
```

```
var_dump(strrpos('ABCDEF','DAF')); //bool(false)
?>
```

Un objet sans propriété n'est plus considéré comme vide (empty)

```
<?php
class test { }
$t = new test();

var_dump(empty($t)); // echo bool(false)

if ($t) {
    // Cette portion de code sera exécutée
}
?>
```

Dans certains cas, les classes doivent être déclarées avant d'être utilisées

```
<?php

// fonctionne sans erreur :
$a = new a();
class a {
}

// ici, une erreur :
$a = new b();

interface c{
}
class b implements c {
}
?>
```

11.2.3 CLI et CGI

PHP 5 introduit des modifications dans les noms des fichiers CLI et CGI. Avec PHP 5, la version CGI s'appelle maintenant php-cgi.exe (auparavant php.exe) et la version CLI se trouve maintenant dans le répertoire principal de PHP (auparavant cli/php.exe).

Un nouveau mode a également été introduit avec PHP 5 : php-win.exe . Cette version est équivalente à la version CLI sauf qu'elle n'ouvre pas de fenêtre de console DOS au lancement. Ce comportement est similaire à celui de php-gtk.

Avec PHP 5, la version CLI renseignera toujours les variables globales \$argc et \$argv en accord avec toutes les directives du php.ini . Même si vous avez la directive register_argc_argv configurée à Off , cela n'aura pas d'effet sur la version CLI.

Voir aussi la documentation sur [la ligne de commande](#) .

11.2.4 Migration des fichiers de configuration

Les modules ISAPI ayant changé de nom (php4xxx vers php5xxx), il est nécessaire de faire quelques modifications dans vos fichiers de configuration. Des modifications des noms des versions CLI et CGI ont également été faites. Consultez la [section correspondante](#) pour plus d'informations.

La migration du fichier de configuration Apache est très facile. Consultez l'exemple ci-dessous afin de connaître les modifications que vous devez faire :

Migration des fichiers de configuration Apache

```
# changez cette ligne :  
LoadModule php4_module /php/sapi/php4apache2.dll  
  
# en :  
LoadModule php5_module /php/php5apache2.dll
```

Si votre serveur web utilise PHP en mode CGI, il vous faut noter que la version CGI a changé de nom, passant de php.exe à php-cgi.exe . Voici la modification à faire, pour Apache :

Migration des fichiers de configuration Apache, en mode CGI

```
# changez cette ligne :  
Action application/x-httpd-php "/php/php.exe"  
  
# en :  
Action application/x-httpd-php "/php/php-cgi.exe"
```

Si vous utilisez un serveur web autre que Apache, vous devez également changer le nom de fichier du module ISAPI ou du CGI.

11.2.5 Nouvelles fonctions

PHP 5 introduit de nouvelles fonctions. En voici la liste :

Tableaux :

- array_combine : crée un tableau en combinant un tableau pour les clés et un autre tableau pour les valeurs correspondantes.
- array_diff_uassoc : renvoie la différence entre deux tableaux avec la possibilité de vérifier les index par l'appel d'une fonction utilisateur de callback.
- array_udiff : renvoie la différence entre deux tableaux en utilisant une fonction de callback pour la comparaison des valeurs.
- array_udiff_assoc : renvoie la différence entre deux tableaux avec la possibilité de vérifier les index. Les valeurs sont elles comparées grâce à une fonction utilisateur de callback.
- array_udiff_uassoc : renvoie la différence entre deux tableaux avec la possibilité de vérifier les index. Les valeurs sont comparées grâce à une fonction utilisateur de callback, les index sont également comparés grâce à une (autre) fonction utilisateur de callback.
- array_walk_recursive : applique une fonction utilisateur de façon récursive à chaque élément d'un tableau.
- array_uintersect_assoc : calcule l'intersection de deux tableaux avec une vérification des index. Les données sont comparées en utilisant une fonction de callback.
- array_uintersect_uassoc : calcule l'intersection de deux tableaux avec une vérification des index. Les données et les index sont comparés en utilisant des fonctions de callback.
- array_uintersect : calcule l'intersection de deux tableaux. Les données sont comparées en utilisant une fonction de callback.

InterBase :

- ibase_affected_rows : renvoie le nombre de lignes modifiées par la requête précédente.
- ibase_backup : lance une tâche de sauvegarde dans le gestionnaire de services et rend la main immédiatement.
- ibase_commit_ret : permet de valider une transaction sans avoir à la fermer.
- ibase_db_info : renvoie des statistiques sur une base de données.

- ibase_drop_db : supprime une base de données.
- ibase_errcode : renvoie un code d'erreur.
- ibase_free_event_handler : annule un gestionnaire d'événement enregistré.
- ibase_gen_id : incrémente le générateur donné et renvoie sa nouvelle valeur.
- ibase_maintain_db : exécute une commande de maintenance sur le serveur de base de données.
- ibase_name_result : assigne un nom à un jeu de résultats.
- ibase_num_params : renvoie le nombre de paramètres d'une requête préparée.
- ibase_param_info : renvoie des informations sur un paramètre spécifique d'une requête préparée.
- ibase_restore : lance une restauration dans le gestionnaire de services et rend la main immédiatement.
- ibase_rollback_ret : annule la transaction et conserve son contexte.
- ibase_server_info : renvoie des statistiques sur une base de données.
- ibase_service_attach : initie une connexion au gestionnaire de services.
- ibase_service_detach : arrête une connexion au gestionnaire de services.
- ibase_set_event_handler : permet d'enregistrer une fonction de callback appelée lorsque des événements sont postés.
- ibase_wait_event : attend qu'un événement soit posté par la base de données.

iconv :

- iconv_mime_decode : décode un champ d'en-tête MIME.
- iconv_mime_decode_headers : décode plusieurs champs d'en-tête MIME en une seule fois.
- iconv_mime_encode : crée un champ d'en-tête MIME.
- iconv_strlen : renvoie le nombre de caractères de la chaîne.
- iconv_strpos : trouve la position de la première occurrence d'une chaîne donnée dans une autre chaîne.
- iconv_strrpos : trouve la position de la dernière occurrence d'une chaîne donnée dans une autre chaîne.
- iconv_substr : supprime un morceau de chaîne.

Flux :

- stream_copy_to_stream : copie les données d'un flux vers un autre.
- stream_get_line : récupère une ligne d'un flux donné jusqu'au délimiteur fourni.
- stream_socket_accept : accepte une connexion sur un socket créé avec stream_socket_server .
- stream_socket_client : ouvre une connexion socket sur un domaine Unix ou Internet.
- stream_socket_get_name : récupère le nom des sockets locaux ou distants.
- stream_socket_recvfrom : reçoit des données d'un socket, qu'il soit connecté ou non.
- stream_socket_sendto : envoie un message vers un socket, qu'il soit connecté ou non.
- stream_socket_server : crée un socket de domaine Unix ou Internet.

Date et heure :

- idate : formate une date/heure en tant qu'entier.
- date_sunset : renvoie l'heure du coucher du soleil pour un jour et un endroit donnés.
- date_sunrise : renvoie l'heure du lever du soleil pour un jour et un endroit donnés.
- time_nanosleep : marque une pause exprimée en secondes et nanosecondes.

Chaîne de caractères :

- str_split : convertit une chaîne en tableau.

- [strpbrk](#) : recherche dans une chaîne l'un des caractères d'une liste.
- [substr_compare](#) : compare deux chaînes depuis un offset jusqu'à une longueur en caractères, d'une façon sécurisée contre les données binaires et, optionnellement, insensible à la casse.

Autres :

- [convert_uudecode](#) : décode une chaîne au format uuencode .
- [convert_uuencode](#) : encode une chaîne au format uuencode .
- [curl_copy_handle](#) : copie un descripteur cURL avec toutes ses préférences.
- [dba_key_split](#) : sépare une clé sous forme de chaîne en représentation sous forme de tableau.
- [dbase_get_header_info](#) : récupère les informations d'en-tête d'une base de données dBase.
- [dbx_fetch_row](#) : récupère une ligne à partir d'un résultat de requête ayant le marqueur DBX_RESULT_UNBUFFERED.
- [fbsql_set_password](#) : change le mot de passe d'un utilisateur donné.
- [file_put_contents](#) : écrit une chaîne dans un fichier.
- [ftp_alloc](#) : alloue de l'espace pour un fichier à uploader.
- [get_declared_interfaces](#) : renvoie un tableau contenant toutes les interfaces déclarées.
- [get_headers](#) : renvoie tous les en-têtes envoyés par le serveur en réponse à une requête HTTP.
- [headers_list](#) : renvoie la liste des en-têtes de réponse envoyés (ou prêts à être envoyés).
- [http_build_query](#) : renvoie une requête encodée sous forme d'URL.
- [image_type_to_extension](#) : renvoie l'extension de fichier pour le type d'image retourné par [getimagesize](#) , [exif_read_data](#) , [exif_thumbnail](#) , [exif_imagetype](#) .
- [imagefilter](#) : applique un filtre sur une image en utilisant un angle donné.
- [imap_getacl](#) : récupère la valeur de l'ACL pour une boîte aux lettres donnée.
- [ldap_sasl_bind](#) : établit une liaison à un répertoire LDAP en utilisant SASL.
- [mb_list_encodings](#) : retourne un tableau contenant tous les encodages supportés.
- [pcntl_getpriority](#) : renvoie la priorité d'un processus.
- [pcntl_wait](#) : attend ou renvoie le statut d'un processus enfant comme défini par l'appel système `waitpid()` .
- [pg_version](#) : renvoie un tableau contenant les versions du client, du protocole et du serveur PostgreSQL, si disponibles.
- [php_check_syntax](#) : vérifie la syntaxe d'un fichier donné.
- [php_strip_whitespace](#) : renvoie la source en supprimant les commentaires et les espaces.
- [proc_nice](#) : change la priorité du processus courant.
- [pspell_config_data_dir](#) : change l'emplacement des fichiers de langues.
- [pspell_config_dict_dir](#) : change l'emplacement de la liste principale de mots.
- [setrawcookie](#) : envoie un cookie sans encodage url de la valeur.
- [snmp_read_mib](#) : lit et analyse un fichier MIB dans l'arbre actif MIB.
- [sqlite_fetch_column_types](#) : renvoie un tableau contenant les types des colonnes pour une table spécifiée.

Note

L'API de l'extension <u>Tidy</u> a changé du tout au tout.
--

11.2.6 Nouvelles Directives

De nouvelles directives pour le fichier `php.ini` ont été ajoutées avec PHP 5. En voici une liste :

- `mail.force_extra_parameters` : permet de forcer l'ajout du paramètre spécifié en tant que paramètre supplémentaire pour `sendmail`. Ces paramètres prendront la place du cinquième paramètre de la fonction [mail](#) quoi qu'il arrive, même en `safe mode`.

- register_long_arrays : permet/interdit à PHP d'utiliser les tableaux \$HTTP_*_VARS .
- session.hash_function : permet de choisir une fonction de hachage (MD5 ou SHA-1).
- session.hash_bits_per_character : permet de définir le nombre de bits utilisés pour chaque caractère lors de conversions de données binaires en éléments lisibles (de 4 à 6).
- zend.ze1_compatibility_mode : active le mode de compatibilité avec le Zend Engine 1 (PHP 4).

11.2.7 Bases de données

Certaines modifications ont été apportées au support des bases de données (MySQL et SQLite).

PHP 5 n'intègre pas les bibliothèques clientes MySQL par défaut pour des raisons de licence entre autres. Pour plus d'informations, lisez la [FAQ](#) .

Il existe également une nouvelle extension, MySQLi (Improved MySQL) , créée pour fonctionner avec MySQL 4.1 et supérieur.

A partir de PHP 5, l'extension SQLite est intégrée à PHP. SQLite est un moteur de bases de données embarqué, il ne s'agit pas simplement d'une bibliothèque cliente permettant de se connecter à un serveur de bases de données (comme MySQL ou PostgreSQL). La bibliothèque SQLite lit et écrit ses bases de données directement dans des fichiers sur le disque.

11.2.8 Nouveau Modèle Objet

PHP 5 inclut un nouveau modèle objet. Le traitement des objets en PHP a complètement été réécrit pour arriver à de meilleures performances et plus de fonctionnalités. Dans les versions précédentes de PHP, les objets sont traités comme des types primitifs (par exemple les entiers ou les chaînes de caractères). L'inconvénient de cette méthode est que sémantiquement, l'objet en entier était copié lorsqu'une variable était assignée ou passée comme paramètre à une fonction. Dans la nouvelle approche, les objets sont référencés par un pointeur et non pas leur valeur (on peut penser à un pointeur en tant qu'identifiant d'objet).

Beaucoup de développeurs PHP ne se rendent pas compte des caprices lors de la copie dans l'ancien modèle objet et, par conséquent, la majorité des applications PHP devrait fonctionner directement ou avec très peu de modifications.

Le nouveau modèle objet est documenté dans la partie de [référence du langage](#) .

Voir aussi la directive zend.ze1_compatibility_mode pour la compatibilité avec PHP 4.

11.2.9 Rapport d'erreurs

En PHP 5, la nouvelle constante de rapport d'erreurs E_STRICT a été introduite avec comme valeur 2048. Cela permet à PHP, lors de l'exécution, de faire des suggestions sur l'interopérabilité et le suivi de votre code. Ceci vous permet de toujours utiliser les meilleures méthodes de codage et les plus récentes, par exemple les messages stricts vous avertiront sur l'utilisation de fonctions obsolètes.

Note
E_ALL n'inclut pas E_STRICT ; il n'est donc pas activé par défaut.

11.3 Migration de PHP 3 à PHP 4

11.3.1 Ce qui a changé en PHP 4

PHP 4 et le moteur Zend ont significativement amélioré les performances et les possibilités de PHP, tout en assurant une compatibilité ascendante maximale. La plupart des scripts existants écrits en PHP 3 fonctionneront sous PHP 4. La migration de votre code de PHP 3 vers PHP 4 sera beaucoup plus facile que celle de PHP/FI 2 vers 3. Un grand nombre de scripts seront prêts sans modification, mais il est bon que vous connaissiez les quelques différences et que vous testiez vos applications avant d'effectuer le changement de cadre de production. Les indications suivantes vous mettront sur la voie.

11.3.2 Utiliser PHP 3 et PHP 4 simultanément

Les systèmes d'exploitation récents disposent de capacités de versioning et de scoping. Ces fonctionnalités rendent possible l'installation de PHP 3 et PHP 4 comme modules Apache, simultanément.

Ceci a été fait sur les plate-formes suivantes :

- Linux avec les binutils récents (testé avec binutils 2.9.1.25)
- Solaris 2.5 ou plus récent
- FreeBSD (testé avec 3.2, 4)

Pour l'activer, configurez PHP 3 et PHP 4 pour qu'ils utilisent APXS (--with-apxs) et les extensions nécessaires (--enable-versioning). En dehors de cela, toutes les instructions d'installation habituelles s'appliquent. Par exemple :

Configuration pour utiliser simultanément PHP 3 et 4

```
$ ./configure \
--with-apxs=/apache/bin/apxs \
--enable-versioning \
--with-mysql \
--enable-track-vars
```

11.3.3 Migration des fichiers de configuration

Le fichier de configuration global, php3.ini , a été renommé en php.ini .

Pour les fichiers de configuration Apache, il y a eu des modifications plus importantes. Les types MIME reconnus par le module PHP ont été modifiés.

Activation des types MIME PHP 3 et 4

```
application/x-httpd-php3 --> application/x-httpd-php
application/x-httpd-php3-source --> application/x-httpd-php-source
```

Vous pouvez faire fonctionner vos deux versions de PHP avec le même fichier de configuration Apache (suivant la version qui est déjà compilée sur le serveur), en utilisant la syntaxe suivante :

Utilisation du même fichier de configuration Apache pour PHP 3 et 4

```
AddType application/x-httpd-php3 .php3
AddType application/x-httpd-php3-source .php3s
AddType application/x-httpd-php .php
```

```
AddType application/x-httpd-php-source .phps
```

De plus, les noms des directives PHP pour Apache ont aussi été modifiés.

Depuis PHP 4, il n'y a que 4 directives Apache qui se rapportent à PHP :

Directives Apache pour PHP 4

```
php_value [PHP directive name] [value]
php_flag [PHP directive name] [On|Off]
php_admin_value [PHP directive name] [value]
php_admin_flag [PHP directive name] [On|Off]
```

Il y a deux différences entre les options Admin et les autres valeurs :

- Les options d'administration ne peuvent être placées que dans le fichier de configuration général (i.e. httpd.conf).
- Les options standard ne peuvent pas contrôler certaines directives PHP. Par exemple, le safe mode (si vous pouviez modifier les configurations dans le fichier .htaccess, cela annulerait toute la sécurité du safe mode). A l'inverse, les valeurs d'administration peuvent modifier n'importe quelle directive PHP.

Pour rendre le processus de transition plus agréable, PHP 4 est distribué avec des scripts qui convertissent automatiquement votre fichier de configuration Apache et vos fichiers .htaccess pour qu'ils puissent fonctionner aussi bien avec PHP 3 que PHP 4. Ces scripts ne convertissent PAS les lignes concernant les types MIME. Vous devez le faire vous-même.

Pour convertir votre fichier de configuration Apache, exécutez le script apconf-conv.sh (disponible dans le dossier scripts/apache/). Par exemple :

Conversion du fichier de configuration Apache pour PHP 4

```
~/php4/scripts/apache:# ./apconf-conv.sh /usr/local/apache/conf/httpd.conf
```

Votre configuration originale sera sauvegardée dans le fichier httpd.conf.orig.

Pour convertir vos fichiers .htaccess, exécutez le script aphtaccess-conv.sh (disponible dans le dossier scripts/apache/). Par exemple :

Conversion des fichiers .htaccess pour PHP 4

```
~/php4/scripts/apache:# find / -name .htaccess -exec ./aphtaccess-conv.sh {} \;
```

De la même façon, votre ancien fichier .htaccess sera sauvé sous le nom .htaccess.orig.

Les scripts de conversion requièrent l'installation préalable de awk.

11.3.4 Comportement de l'analyseur

L'analyse et l'exécution sont désormais deux étapes complètement dissociées, et l'exécution intervient lorsque le code, avec tous ses inclusions et pré-requis, a été complètement analysé et validé.

Une des nouvelles conditions introduites est que les fichiers inclus et requis (include et require) doivent être syntaxiquement complets. Vous ne pouvez plus répartir différents cas de votre code dans plusieurs fichiers. Vous ne pouvez plus commencer une boucle for ou while, une condition if ou un cas switch dans un fichier, et finir la boucle ou placer les cas else, endif, case ou break dans un autre fichier.

Il est toujours valable d'inclure du code supplémentaire depuis une boucle ou dans une condition, mais les accolades de bloc {...} , et les éléments de la boucle doivent être dans le même fichier ou dans la même chaîne évaluée avec eval .

Cela ne devrait pas perturber trop de monde car éclater son code ainsi est de toute façon considéré comme un très mauvais style de programmation.

Une autre nouveauté est qu'il n'est plus possible de faire retourner une valeur avec un fichier requis (require) (mais c'est plutôt rare en PHP 3). Retourner une valeur avec un fichier inclus (include) est toujours possible.

11.3.5 Rapport d'erreur

11.3.5.1 Changement de configuration

Avec PHP 3, le niveau de rapport d'erreur était obtenu en ajoutant les constantes numériques de chaque niveau de rapport. Généralement, on utilisait 15 pour afficher toutes les erreurs, et 7 pour afficher toutes les erreurs hormis les alertes simples.

PHP 4 dispose d'un nombre significativement plus grand de niveaux de rapport d'erreur, et l'analyseur accepte désormais les constantes symboliques destinées à configurer le comportement désiré.

Le niveau de rapport d'erreur doit désormais être explicitement configuré en supprimant les niveaux dont vous ne voulez pas, grâce à la fonction de OU exclusif. Ça a l'air compliqué? Supposons que vous souhaitiez afficher toutes les erreurs, hormis les alertes de style, qui sont repérées par la constante : E_NOTICE. Il suffit d'ajouter la valeur suivante dans le fichier php.ini : error_reporting = E_ALL & ~E_NOTICE . Si vous voulez supprimer aussi les alertes, vous pouvez ajouter la constante appropriée, en la combinant avec l'opérateur OU logique '|': error_reporting= E_ALL & ~(E_NOTICE | E_WARNING) .

Attention
Lors de la mise à jour de votre code ou de vos serveurs de PHP 3 à PHP 4, vous devez vérifier ces valeurs de configuration et appeler la fonction <u>error_reporting</u> ou bien désactiver les nouveaux types d'erreurs, tout spécialement E_COMPILE_ERROR . Ceci peut mener à des documents vides sans aucune possibilité de tracer ce qui s'est produit ou où rechercher le problème.
Attention
L'utilisation des anciennes valeurs 7 et 15 est une très mauvaise idée, car elles ne prennent pas en compte les nouvelles classes d'erreurs, y compris certaines erreurs d'analyse. Cela peut conduire à des résultats très étranges, où le script n'affiche plus rien, malgré une erreur d'analyse.
Cela a conduit dans le passé à un grand nombre de rapports de bogues sur l'analyseur, alors que les programmeurs n'étaient tout simplement pas capables de repérer l'accolade manquante dans un fichier requis, car l'analyseur avait la consigne de cacher ces erreurs.
Vérifier votre niveau d'erreur doit être le premier réflexe lorsque vos scripts meurent silencieusement. Le moteur Zend est considéré actuellement comme suffisamment mature pour ne plus causer ce genre de problème aujourd'hui.

11.3.5.2 Nouveaux messages d'erreurs

Un grand nombre de scripts PHP 3 utilisent des structures qui doivent être considérées comme un très mauvais style, même si elles effectuent bien les tâches qui leur sont affectées, car ils ne sont pas robustes. PHP 4 affichera de nombreux messages d'erreur dans des situations où PHP 3 restera coi. La solution de facilité consiste à supprimer les messages de niveau E_NOTICE, mais c'est une meilleure idée de plutôt corriger le code.

Le cas le plus courant qui générera des messages d'alertes est l'utilisation de constantes sans guillemets comme index de tableaux. PHP 3, comme PHP 4, finiront par les interpréter littéralement comme des chaînes, si aucune constante n'est définie à la place. Mais si jamais une telle constante est définie dans une autre partie du code, cela risque de produire des résultats étonnants. Cela peut devenir un trou de sécurité si un pirate arrive à redéfinir les constantes de telle manière que le script lui donne accès à un niveau de droits supérieur. PHP 4 vous signalera tout oubli de guillemets par exemple dans : `$_SERVER[REQUEST_METHOD]` . Modifier ce code en `$_SERVER['REQUEST_METHOD']` rendra l'analyseur heureux, et améliorera grandement votre style et la sécurité du code.

PHP 4 vous signalera les variables ou les éléments de tableaux non initialisés.

11.3.6 Initialiseur

Les variables statiques et les membres de classes n'acceptent plus que des initialiseurs scalaires, tandis que PHP 3 acceptait aussi les expressions. Ceci est dû, encore une fois, à la séparation de l'analyse et de l'exécution : aucun code ne peut être exécuté tant que l'analyse n'est pas terminée.

Pour les classes, il vaut mieux initialiser les membres dans le constructeur. Pour les variables statiques, une valeur fixe et simple est la seule chose qui vienne à l'esprit.

11.3.7 `empty("0")`

L'évolution la plus polémique est celle de `empty` . Une chaîne contenant seulement le caractère '0' (zéro) est maintenant considérée comme vide, alors qu'elle ne l'était pas en PHP 3.

Ce nouveau comportement prend tout son sens dans les applications web, puisque tous les résultats de champs de type input sont de type chaîne de caractères, même si un nombre est demandé, et ce, grâce aux capacités de conversion automatique de PHP. D'un autre côté, cela peut casser votre code d'une manière très subtile, menant droit au comportement erratique, difficilement repérable si vous ne savez pas ce qui vous attend.

11.3.8 Fonctions manquantes

Bien que PHP 4 dispose de nombreuses fonctionnalités, fonctions et extensions nouvelles, vous pouvez trouver des fonctions PHP 3 qui manquent. Un petit nombre de fonctions de base n'ont pu être portées en PHP 4, maintenant que l'analyse et l'exécution ont été séparées. D'autres fonctions, et même des extensions entières sont maintenant obsolètes, remplacées par de nouvelles fonctions plus puissantes ou plus efficaces. Certaines fonctions n'ont tout simplement pas été portées pour le moment ou pour des raisons de licence.

11.3.8.1 Fonctions manquantes pour des raisons de structure

Comme PHP 4 sépare l'analyse et l'exécution, il n'est plus possible de modifier le comportement de l'analyseur (intégré dans le moteur Zend) durant l'exécution, puisque toute l'analyse a eu lieu, et est terminée. La fonction `short_tags()` a cessé d'exister. Vous pouvez toujours modifier le comportement de l'analyseur avec le fichier `php.ini`.

Une autre fonctionnalité de PHP 3 qui ne fait pas partie de PHP 4 est le débogueur. Des débogueurs réalisés par des tiers sont disponibles, mais ne sont pas distribués avec PHP. D'autres extensions du Zend Engine sont aussi disponibles auprès de divers sites et groupes.

11.3.8.2 Fonctions et extensions obsolètes

Les extensions Adabas et Solid n'existent plus. Longue vie à l'extension ODBC unifiée à la place!

11.3.8.3 Nouveau statut pour `unset`

`unset`, bien que toujours disponible, a été implémentée comme une structure du langage et elle n'est plus vraiment une "fonction".

Cela n'a pas de conséquence directe sur le comportement de `unset`, mais tester son existence avec `function_exists` retournera `FALSE` comme cela serait le cas avec une autre structure du langage comme `echo`.

Une autre application pratique disparue est qu'il n'est plus possible d'appeler `unset` indirectement, c'est-à-dire que `$func = "unset"; $func($somevar)` ne fonctionnera plus.

11.3.9 Extensions PHP 3

Les extensions écrites pour PHP 3 ne fonctionnent plus avec PHP 4, ni les exécutables, ni les codes source. Il n'est pas difficile de porter les extensions de PHP 3 à 4 si vous avez accès aux sources originales. Cet appendice ne contient pas de description détaillée de ce processus de portage.

11.3.10 Substitution de variables dans les chaînes

PHP 4 dispose d'un nouveau mécanisme de substitution des variables dans les chaînes. Vous pouvez désormais accéder aux membres d'objets et aux tableaux multidimensionnels dans une chaîne.

Pour ce faire, il suffit de placer la variable entre accolades, le signe `$` suivant immédiatement la première accolade : `{ $variable['a'] }`

Pour utiliser la valeur d'un membre d'objet dans une chaîne, il suffit d'écrire : `"texte { $objet->membre } texte"` ; alors qu'en PHP 3, il fallait faire comme ceci : `"texte " . $objet->membre . " texte"`.

Cette technique rend le code beaucoup plus lisible, mais risque de poser des problèmes dans certains scripts PHP 3. Vous pouvez facilement traquer ce problème en recherchant les séquences `{ $` dans votre code, et en les remplaçant par `\{ $` avec votre outil de remplacement préféré.

11.3.11 Cookies

PHP 3 avait la mauvaise habitude d'envoyer les cookies dans l'ordre inverse de celui des appels à `setcookie` dans le code. PHP 4 rompt avec cette habitude en les envoyant dans le même ordre que vous les mettez dans le code.

Cela peut aussi prendre à contre-pied certains programmes, mais l'ancien comportement était tellement étrange qu'il méritait un tel traitement un jour ou l'autre, pour éviter d'autres problèmes ultérieurs.

11.3.12 Gestion des variables globales

Alors que la gestion des variables globales a été surtout étudiée pour être simple en PHP 3 et dans les premières versions de PHP 4, la sécurité est devenue la priorité avec les dernières versions. Alors qu'en PHP 3 l'exemple ci-dessous fonctionnait sans problème, en PHP 4 il faut utiliser `unset($GLOBALS["id"]);`. Ceci n'est qu'un des problèmes liés à la gestion des variables globales. Vous auriez toujours dû utiliser `$GLOBALS`, avec les nouvelles versions de PHP vous y êtes obligés dans la plupart des cas. Apprenez-en plus sur ce sujet dans la section [références globales](#).

Migration des variables globales

```
<?php
    $id = 1;
    function test() {
        global $id;
        unset($id);
    }
    test();
    echo($id);
    // Ceci va afficher 1 en PHP 4
?>
```

11.4 Migration de PHP/FI 2 à PHP 3

11.4.1 A propos des incompatibilités en 3.0

PHP 3.0 a été réécrit depuis le début. Le nouvel analyseur syntaxique est beaucoup plus robuste et cohérent qu'en version 2.0. Il est aussi nettement plus rapide et utilise encore moins de mémoire. Cependant, ces améliorations n'ont pu être possibles qu'au prix de modifications parfois importantes, tant au niveau de la syntaxe qu'au niveau des fonctionnalités.

De plus, l'équipe de développement PHP a essayé de nettoyer la syntaxe et la sémantique, ce qui a aussi causé quelques incompatibilités. A long terme, nous pensons que ces modifications seront pour le bien de tous.

Ce chapitre va tenter de vous montrer les incompatibilités que vous pourriez rencontrer lors de votre migration de PHP/FI 2.0 à PHP 3.0 et de vous aider à les résoudre. Les nouvelles fonctionnalités ne sont pas signalées, à moins que cela ne soit nécessaire.

Un programme de conversion automatique de vos anciens scripts PHP/FI 2.0 existe. Il est disponible dans le dossier de convertisseur de la distribution PHP 3.0: c'est le dossier `convertor`. Ce programme ne fait que repérer les modifications de syntaxe et ne vous épargnera pas une relecture attentive du script.

11.4.2 old_function

L'instruction `old_function` vous permet de déclarer une fonction en utilisant une syntaxe identique à celle de PHP/FI (hormis le fait que vous devez remplacer " fonction " par " `old_function` ").

C'est une fonctionnalité obsolète, qui ne doit être utilisée que par les convertisseurs PHP/FI 2 -> PHP 3.

Attention

Les fonctions déclarées comme `old_function` ne peuvent pas être appelées depuis le code interne PHP. Entre autres choses, cela signifie que vous ne pouvez pas les utiliser avec des fonctions comme `usort` , `array_walk` et `register_shutdown_function` . Vous pouvez contourner cette limitation en écrivant une fonction moderne, qui appelle cette vieille syntaxe `old_function` .

11.4.3 Balises PHP

La première chose que vous remarquerez probablement est que les balises ouvrantes et fermantes de PHP ont changé. L'ancienne forme `<? >` a été remplacée par trois nouvelles balises possibles :

Migration : balises PHP

```
<?
echo "Ceci est du code PHP/FI 2.0.\n";
>
```

Comme en version 2.0, PHP/FI accepte aussi cette variante :

Migration : premières nouvelles balises PHP

```
<?
echo "Ceci est du code PHP 3.0!\n";
?>
```

Notez bien que la balise de fin contient désormais un point d'interrogation et un signe supérieur ">". Cependant, si vous souhaitez utiliser XML sur votre serveur, vous aurez sûrement des problèmes avec cette variante, car PHP risque d'essayer d'exécuter des balises XML. A cause de ceci, la notation suivante a été ajoutée :

Migration : nouvelles balises PHP

```
<?php
echo "Ceci est du code PHP 3.0!\n";
?>
```

Certains d'entre vous rencontrent des problèmes avec les éditeurs qui ne comprennent pas ce type de balises d'instruction : Microsoft FrontPage est l'un de ces éditeurs, et, pour contourner le problème, la variation suivante a été introduite :

Nouvelles balises PHP

```
<script language="php">
    echo "Ceci est du code PHP 3.0!\n";
</script>
```

11.4.4 Syntaxe if...endif

La syntaxe alternative pour écrire des instructions `if/elseif/else` , avec `if()`; `elseif()`; `else`; `endif`; ne pouvait pas être conservée sans ajouter beaucoup de complexité à l'analyseur syntaxique. De ce fait, cette syntaxe a été modifiée :

Migration : ancienne syntaxe if...endif

```
<?php
if ($foo);
    echo "oui\n";
elseif ($bar);
    echo "presque\n";
else;
    echo "non\n";
endif;
?>
```

Migration : nouvelle syntaxe if...endif

```
<?php
if ($foo):
    echo "oui\n";
elseif ($bar):
    echo "presque\n";
else:
    echo "non\n";
endif;
?>
```

Notez que les points virgules ont été remplacée par des points dans toutes les commandes, sauf pour la dernière expression (endif).

11.4.5 Syntaxe while

Tout comme pour if...endif , la syntaxe des boucles while...endwhile a changée :

Migration : ancienne syntaxe while...endwhile

```
<?php
while ($more_to_come);
    //...
endwhile;
?>
```

Migration : nouvelle syntaxe while...endwhile

```
<?php
while ($more_to_come):
    ...
endwhile;
?>
```

Attention

Attention : si vous utilisez la vieille syntaxe while...endwhile en PHP 3.0, vous obtiendrez une boucle sans fin !

11.4.6 Types d'expression

PHP/FI 2.0 utilisait le membre à gauche dans les expressions, pour déterminer le type de résultat attendu. PHP 3.0 prend en compte les deux côtés de l'expression et cela peut produire des résultats inattendus avec les scripts 2.0.

Considérez les lignes suivantes :

Différence entre PHP 2 et PHP 3

```
<?php
$a[0]=5;
$a[1]=7;
$key = key($a);
while (" " != $key) {
    echo "$keyn";
    next($a);
}
?>
```

En PHP/FI 2.0, cet exemple va afficher les indices des \$a . En PHP 3.0, l'exemple ne va rien afficher du tout. La raison est qu'en PHP 2.0, puisque l'argument de gauche est de type chaîne, une comparaison de chaîne était effectuée et, effectivement, "" n'est pas "" , ce qui conduit la boucle à continuer. En PHP 3, lorsqu'une chaîne est comparée avec un entier, la comparaison est de type chaîne (la chaîne est convertie en entier). Ce qui revient à faire la comparaison entre (atoi("")) qui vaut 0 et la variable qui vaut aussi 0 et comme 0==0, la boucle ne commence même pas.

La correction de ceci est simple. Il suffit de remplacer les commandes while par :

Nouvelle syntaxe PHP3

```
<?php
while ((string)$key != "") {
?>
```

11.4.7 Les messages d'erreur ont changé

Les messages d'erreur en PHP 3.0 sont généralement plus précis que ceux de la version 2.0., mais vous ne verrez plus la portion de code qui a causé l'erreur. A la place, un numéro de ligne et un nom de fichier sera retourné.

11.4.8 Evaluation rapide des booléens

En PHP 3, l'évaluation des est court-circuité. Cela signifie dans une expression telle que ((1 || test_me())), la fonction test_me() ne sera pas exécutée, car cela ne changera pas le résultat.

C'est une amélioration mineure, mais qui peut avoir des effets secondaires importants.

11.4.9 La valeur TRUE / FALSE comme retour de fonctions

La plupart des fonctions internes de PHP ont été réécrite pour qu'elle retourne TRUE en cas de succès, et FALSE en cas d'erreur, au contraire des fonctions qui retournaient 0 et -1 en PHP/FI 2.0. Le nouveau comportement est beaucoup plus logique, comme \$fp = fopen("/your/file") or fail("fichier non trouvé!"); . Etant donné que PHP/FI 2.0 n'a pas de règle claire à propos de ce que les fonctions doivent retourner en cas d'échec, la plupart des scripts devront probablement être vérifié manuellement, après avoir utilisé le convertisseur 2.0 à 3.0.

Migration depuis 2.0 : valeur retournées, ancienne méthode

```
<?php
$fp = fopen($file, "r");
if ($fp == -1);
    echo("Impossible d'ouvrir le fichier $file en lecture <br />\n");
endif;
?>
```

Migration depuis 2.0 : valeur retournées, nouvelle façon

```
<?php
$fp = @fopen($file, "r") or
print("Impossible d'ouvrir le fichier $file en lecture<br />\n");
?>
```

11.4.10 Diverses incompatibilités

- Le module PHP 3.0 pour Apache n'accepte plus les versions d'Apache antérieure à la version 1.2. Apache 1.2 ou plus récent est nécessaire.
- `echo` n'utilise plus de chaîne de formatage. Il faut utiliser `printf` à la place.
- En PHP/FI 2.0, un effet secondaire de l'implémentation faisait que `$foo[0]` était la même chose que `$foo`. Ce n'est plus vrai en PHP 3.0.
- Lire un tableau avec `$array[]` n'est plus valable.

Ainsi, il n'est plus possible de passer en revue un tableau avec des boucles telles que `$data = $array[]`. Utilisez `current` et `next` à la place.

Ainsi, `$array1[] = $array2` n'ajoute pas les valeurs de `$array2` à `$array1`, mais crée un nouvel élément dans `$array1` et y affecte `$array2` comme dernier élément. Voir aussi les tableaux multidimensionnels.

- "+" n'est plus utilisable comme opérateur de concaténation de chaîne. A la place, il convertit les arguments en nombres et effectue une addition numérique. Utilisez "." à la place.

Migration depuis 2.0 : concaténation de chaînes

```
<?php
echo "1" + "1";
?>
```

En PHP 2.0 cela retournerait 11, en PHP 3.0 cela va retourner 2. A la place, faites :

```
<?php
echo "1"."1";
?>
```

```
<?php
$a = 1;
$b = 1;
echo $a + $b;
?>
```

Cela va afficher 2, tant en PHP 2.0 qu'en 3.0.

```
<?php
$a = 1;
$b = 1;
echo $a.$b;
?>
```

Cela va afficher 11 en PHP 3.0.

11.5 Débugueur PHP

11.5.1 A propos du débogueur

PHP 3 inclut le support d'un serveur de débogage.

PHP 4 n'inclut aucun support de débogage. Vous pouvez toutefois utiliser un des débogueur externe. Le [Studio Zend](#) inclut un débogueur et il y a aussi des débogueurs gratuits comme <http://dd.cron.ru/dbg/> , le [Advanced PHP Debugger](#) (APD) ou [Xdebug](#) qui a toujours une interface de débogage compatible avec les fonctionnalités de débogage de PHP 3 comme décrit dans cette section.

11.5.2 Utiliser le débogueur PHP

Le débogueur PHP sert à repérer les bogues récalcitrants. Le débogueur fonctionne en se connectant à un port TCP à chaque démarrage de PHP. Tous les messages d'erreur seront envoyés sur cette connexion. Cette page est faite pour un "serveur de débogage", qui peut fonctionner avec un IDE ou un éditeur programmable (tel que Emacs).

Comment paramétrer le débogueur :

1. Réservez un port TCP pour le débogueur dans le fichier [de configuration](#) ([debugger.port](#)) et activez-le ([debugger.enabled](#)).
2. Configurer un client TCP sur ce port (par exemple `socket -l -s 1400` sous UNIX).
3. Dans votre code, placez la ligne `"debugger_on(host)"`, où `host` est l'IP ou le nom de l'hôte qui supporte le client TCP .

Désormais, toutes les alertes, notes, etc. seront envoyées sur la socket client, **même si vous avez désactivé le rapport d'erreur avec [error_reporting](#)** .

11.5.3 Protocole du débogueur

Le protocole de débogage PHP 3 fonctionne ligne par ligne. Chaque ligne a un type **type** et plusieurs lignes composent un message Chaque message commence avec une ligne du type `start` et se termine avec une ligne de type `end` . PHP peut envoyer des lignes de plusieurs messages simultanément.

Voici un exemple de ligne à ce format :

exemple de ligne au format de débogage	
<code>date</code>	<code>time</code>
<code>host(pid)</code>	<code>type: message-data</code>

`date`

Les dates sont au format ISO 8601 (yyyy - mm - dd)

`time`

Les heures, y compris les micro-secondes : hh : mm : uuuuuu

`host`

Le nom DNS ou adresse IP de l'hôte qui a généré l'erreur.

`pid`

PID (process id) sur l'hôte `host` , qui a généré l'erreur.

type

Type de la ligne. Indique au programme client comment traiter les données suivantes :

Nom	Signification
start	Indique au programme client que le message du débogueur commence ici. Le contenu de data sera un type d'erreur, comme listé ci-dessous.
message	Le message d'erreur PHP 3.
location	Nom du fichier et numéro de ligne où l'erreur est survenue. La première occurrence de location contiendra toujours la localisation générale. data contiendra : file : line . Il y a toujours une indication de location après un message et après chaque fonction .
frames	Nombre de frames dans le dump de la pile. S'il y a 4 frames, attendez-vous à des informations sur 4 niveaux de fonctions. Si la ligne "frame" n'existe pas, la profondeur doit être 0 (une erreur est survenue au niveau général).
function	Nom de la fonction qui a généré l'erreur. Elle sera répétée à chaque niveau de la pile d'appel.
end	Indique au client que le message du débogueur se termine ici.

data

Ligne de données.

Débogueur	Interne PHP 3
alerte (warning)	E_WARNING
erreur	E_ERROR
analyse (parse)	E_PARSE
note (notice)	E_NOTICE
core-error	E_CORE_ERROR
core-warning	E_CORE_WARNING
inconnue	(toutes les autres)
Exemple de message du débogueur	
<pre>1998-04-05 23:27:400966 lucifer.guardian.no(20481) start: notice 1998-04-05 23:27:400966 lucifer.guardian.no(20481) message: Uninitialized variable 1998-04-05 23:27:400966 lucifer.guardian.no(20481) location: (null):7 1998-04-05 23:27:400966 lucifer.guardian.no(20481) frames: 1 1998-04-05 23:27:400966 lucifer.guardian.no(20481) function: display 1998-04-05 23:27:400966 lucifer.guardian.no(20481) location: /home/ssb/public_html/test.php3:10 1998-04-05 23:27:400966 lucifer.guardian.no(20481) end: notice</pre>	

11.6 Options de configuration

11.6.1 Listes des options de configuration internes

Ci-dessous une liste partielle des options de configuration utilisées par le script PHP configure lors de la compilation sur les systèmes Unix et assimilés. La plus part des options de configuration sont listées dans leur emplacement approprié sur les pages de référence de l'extension et non pas ici. Pour une liste complète des options de configuration, exécutez la commande `./configure --help` dans le répertoire contenant les sources de PHP après avoir exécuté la commande `autoconf` (voir aussi le chapitre sur [l'installation de PHP](#)). Vous pouvez également être intéressé par la lecture de la documentation sur la [configuration GNU](#) pour plus d'informations sur les options de la commande `configure` comme, par exemple, `--prefix=PREFIX` .

Note

Ils sont également utilisés lors de la compilation. Si vous voulez modifier la configuration de l'exécution de PHP, lisez le chapitre sur la [configuration de l'exécution](#) .

- [Divers](#)
- [Comportement de PHP](#)
- [Serveur](#)

11.6.1.1 Options de configuration dans PHP 4

Note

Ces options sont uniquement utilisable en PHP 4 depuis la version 4.1.0. Quelques unes sont valables dans les versions antérieures à PHP 4, quelques unes depuis PHP 3, quelques unes uniquement en PHP 4.1.0. Si vous voulez compiler des anciennes versions, certaines options ne seront probablement pas disponibles.

11.6.1.1.1 Options diverses

- enable-debug
Compile en activant les symboles de débogage
- with-layout=TYPE
Spécifie la façon dont les fichiers installés seront présentés. TYPE peut valoir PHP (valeur par défaut) ou GNU .
- with-pear=DIR
Installe PEAR dans le dossier DIR (par défaut, PREFIX/lib/php).
- without-pear
Ne pas installer PEAR.
- enable-sigchild
Active le propre traitement du SIGCHLD de PHP.
- disable-rpath
Désactive le passage de chemins additionnels de recherche de bibliothèques d'exécution.
- enable-libgcc
Active explicitement le lien avec la bibliothèque libgcc.
- enable-php-streams
Inclut le support expérimental des flux PHP. Ne l'utiliser seulement si vous testez le code !
- with-zlib-dir[=DIR]
Définit le chemin du répertoire d'installation de la bibliothèque zlib.
- enable-trans-sid
Active la propagation transparente de l'identifiant de session. Seulement valable pour PHP 4.1.2 et antérieures. A partir de PHP 4.2.0, cette fonctionnalité est toujours compilée.
- with-tsrm-pthreads
Utilise les threads POSIX (par défaut).
- enable-shared[=PKGS]
Compile les bibliothèques de partage [default=oui].
- enable-static[=PKGS]
Compile les bibliothèques statiques [default=oui].
- enable-fast-install[=PKGS]
Optimise pour une installation rapide [default=oui].
- with-gnu-ld
Le compilateur C utilise GNU ld [default=non].

- disable-libtool-lock
Évite de verrouiller (cela pourrait casser des constructions parallèles).
- with-pic
Tente d'utiliser uniquement les objets PIC/non-PIC [default=utilisation des deux].
- enable-memory-limit
Compile avec le support de limitation de mémoire.
- disable-url-fopen-wrapper
Désactive le wrapper fopen pour les URLs qui permet d'accéder à des fichiers via HTTP ou FTP.
- enable-versioning
Exporte uniquement les symboles requis. Voir le fichier INSTALL pour plus d'informations.
- with-imsp[=DIR]
Inclut le support IMSp (DIR est le répertoire d'inclusion IMPs et le répertoire libimsp.a). PHP 3 uniquement!
- with-mck[=DIR]
Inclut le support Cybercash MCK. DIR est le répertoire d'installation Cybercash MCK, qui vaut par défaut /usr/src/mck-3.2.0.3-linux ; l'aide est disponible dans extra/cyberlib. Uniquement en PHP 3.
- with-mod-dav=DIR
Inclut le support DAV via le module Apache mod_dav, DIR est le répertoire d'installation de mod_dav (version Apache module uniquement !) PHP 3 uniquement !
- enable-debugger
Compile avec les fonctions de déboguages. PHP 3 uniquement !
- enable-versioning
Tire profit du versioning et des portées fournis par Solaris 2.x et Linux. PHP 3 uniquement !

11.6.1.1.2 Options PHP

- enable-maintainer-mode
Active des règles de compilation (make) et des dépendances pas toujours utiles aux utilisateurs occasionnels.
- with-config-file-path=PATH
Indique le chemin dans lequel réside le fichier php.ini . Par défaut, c'est PREFIX/lib .
- enable-safe-mode
Active le safe mode (mode sécurisé) par défaut.
- with-exec-dir[=DIR]
Autorise uniquement des exécutables dans le dossier DIR lorsque le safe mode est activé ; par défaut, DIR vaut /usr/local/php/bin.
- enable-magic-quotes
Active les guillemets magiques.
- disable-short-tags
Désactive les balises courtes <? .

11.6.1.1.3 Options SAPI

La liste suivante contient les options valides SAPI&s (Server Application Programming Interface) pour PHP.

- with-aolserver=DIR
Spécifie le chemin d'installation du serveur AOLserver.

- `--with-apxs[=FILE]`
Compile un module Apache partagé. FILE est un chemin d'accès optionnel vers les outils apxs d'Apache. Par défaut, c'est apxs. Assurez-vous de spécifier la version d'apxs qui est réellement installée sur votre système, et NON pas celle qui est fournie avec Apache.
- `--with-apache[=DIR]`
Compile le module Apache. DIR est le chemin du dossier d'installation d'Apache. Par défaut, DIR vaut /usr/local/apache .
- `--with-mod_charset`
Active les tables de transfert pour mod_charset (version russe de Apache).
- `--with-apxs2[=FILE]`
Compile le module partagé Apache 2.0. FILE est un chemin d'accès optionnel vers les outils apxs d'Apache. Par défaut, c'est apxs.
- `--with-caudium=DIR`
Compile PHP comme module Pike pour utilisation avec Caudium. DIR est le serveur Caudium. Par défaut, c'est /usr/local/caudium/server .
- `--disable-cli`
Disponible depuis PHP 4.3.0. Empêche la compilation de la version CLI de PHP (cela force aussi l'option `--without-pear`). Disponible depuis PHP 4.3.0. Plus d'informations dans la section sur [l'utilisation de PHP en ligne de commande](#) .
- `--enable-embed[=TYPE]`
Active la compilation de la bibliothèque intégrée SAPI. TYPE vaut soit shared , soit static . Par défaut, c'est shared . Disponible depuis PHP 4.3.0.
- `--with-fhttpd[=DIR]`
Compile le module fhttpd. DIR est le dossier d'installation de fhttpd. Par défaut, DIR vaut /usr/local/src/fhttpd . Disponible depuis PHP 4.3.0.
- `--with-isapi=DIR`
Compile PHP sous forme de module ISAPI pour utiliser avec le serveur Zeus.
- `--with-nsapi=DIR`
Spécifie le chemin vers le serveur web Netscape/iPlanet/SunONE.
- `--with-phttpd=DIR`
Aucune information fournie actuellement.
- `--with-pi3web=DIR`
Compile PHP sous forme de module pour utiliser avec le serveur Pi3Web.
- `--with-roxen=DIR`
Compile PHP sous forme de module Pike. DIR est le chemin d'installation de Roxen. Par défaut, DIR vaut /usr/local/roxen/server .
- `--enable-roxen-zts`
Compile PHP sous forme de module Roxen, en utilisant Zend Thread Safety.
- `--with-servlet[=DIR]`
Inclut le support des servlets. DIR est le chemin d'installation du JSDK. Cette SAPI impose que l'extension java soit compilée comme une bibliothèque partagée.
- `--with-thttpd=SRCDIR`
Compile PHP comme module thttpd.
- `--with-tux=MODULEDIR`
Compile PHP comme module TUX (Linux uniquement).
- `--with-webjames=SRCDIR`
Build PHP as a WebJames module (RISC OS only)
- `--disable-cgi`
Empêche la compilation de la version CGI de PHP. Disponible depuis PHP 4.3.0.
- `--enable-force-cgi-redirect`
Active la sécurité lors des redirections internes du serveur. Il est recommandé d'utiliser cette option lorsque vous utilisez PHP en CGI avec Apache.
- `--enable-discard-path`

Si cette option est activée, l'exécutable CGI PHP peut être placé hors de l'arborescence web, en toute sécurité. Il ne sera pas possible de contourner les fichiers .htaccess . security.

--with-fastcgi

Compile PHP comme application FastCGI. Cette option n'est plus disponible depuis PHP 4.3.0 et vous devez utiliser --enable-fastcgi à la place.

--enable-fastcgi

Si cette option est activée, le module CGI sera compilé avec le support FastCGI. Cette option est disponible depuis PHP 4.3.0.

--disable-path-info-check

Si cette option est désactivée, des chemins tels que /info.php/test?a=b ne pourront pas fonctionner. Cette option est disponible depuis PHP 4.3.0. Pour plus d'informations, voir le [Manuel Apache](#) .

11.7 Directives du php.ini

11.7.1 Liste des directives du php.ini

Cette liste inclut les directives du php.ini que vous pouvez définir pour configurer votre installation de PHP.

Nom	Par défaut	Modifiable	Historique
allow_call_time_pass_reference	"1"	PHP_INI_PERDIR	PHP_INI_PERDIR PHP <= 4.3.0
allow_url_fopen	"1"	PHP_INI_SYSTEM	PHP_INI_SYSTEM PHP <= 4.3.0 Disponi PHP 4.0
allow_url_include	"0"	PHP_INI_SYSTEM	Disponi PHP 5.0
always_populate_raw_post_data	"0"	PHP_INI_PERDIR	PHP_INI_PERDIR PHP <= 4.3.0 Disponi PHP 4.0
apc.cache_by_default	"1"	PHP_INI_SYSTEM	
apc.dynamic_error	"0"	PHP_INI_SYSTEM	
apc.enabled	"1"	PHP_INI_ALL	
apc.enable_cli	"0"	PHP_INI_SYSTEM	
apc.file_update_protection	"2"	PHP_INI_SYSTEM	
apc.filters	NULL	PHP_INI_SYSTEM	
apc.gc_ttl	"3600"	PHP_INI_SYSTEM	
apc.max_file_size	"1M"	PHP_INI_SYSTEM	
apc.mmap_file_mask	NULL	PHP_INI_SYSTEM	
apc.num_files_hint	"1000"	PHP_INI_SYSTEM	
apc.optimization	"0"	PHP_INI_ALL	
apc.shm_segments	"1"	PHP_INI_SYSTEM	
apc.shm_size	"30"	PHP_INI_SYSTEM	
apc.slam_defense	"0"	PHP_INI_SYSTEM	

<u>apc.ttl</u>	"0"	PHP_INI_SYSTEM	
apc.user_entries_hint	"100"	PHP_INI_SYSTEM	
apc.user_ttl	"0"	PHP_INI_SYSTEM	
<u>apd.dumpdir</u>	NULL	PHP_INI_ALL	
apd.statement_tracing	"0"	PHP_INI_ALL	
<u>arg_separator.input</u>	"&"	PHP_INI_PERDIR	Disponible PHP 4.0
<u>arg_separator.output</u>	"&"	PHP_INI_ALL	Disponible PHP 4.0
<u>asp_tags</u>	"0"	PHP_INI_PERDIR	PHP_INI_PERDIR PHP <=
<u>assert.active</u>	"1"	PHP_INI_ALL	
<u>assert.bail</u>	"0"	PHP_INI_ALL	
<u>assert.callback</u>	NULL	PHP_INI_ALL	
<u>assert.quiet_eval</u>	"0"	PHP_INI_ALL	
<u>assert.warning</u>	"1"	PHP_INI_ALL	
<u>auto_append_file</u>	NULL	PHP_INI_PERDIR	PHP_INI_PERDIR PHP <=
<u>auto_detect_line_endings</u>	"0"	PHP_INI_ALL	Disponible PHP 4.3
<u>auto_globals_jit</u>	"1"	PHP_INI_PERDIR	Disponible PHP 5.0
<u>auto_prepend_file</u>	NULL	PHP_INI_PERDIR	PHP_INI_PERDIR PHP <=
<u>bcmath.scale</u>	"0"	PHP_INI_ALL	Disponible PHP 5.0
bcompiler.enabled	"1"	PHP_INI_ALL	
birdstep.max_links	"-1"	PHP_INI_ALL	
blenc.key_file	"/usr/local/etc/blenckeys"	PHP_INI_ALL	
<u>browscap</u>	NULL	PHP_INI_SYSTEM	
<u>cgi.fix_pathinfo</u>	"1"	PHP_INI_ALL	
<u>cgi.force_redirect</u>	"1"	PHP_INI_ALL	
cgi.nph	"0"	PHP_INI_ALL	
<u>cgi.redirect_status_env</u>	NULL	PHP_INI_ALL	
<u>child_terminate</u>	"0"	PHP_INI_ALL	Disponible PHP 4.0
<u>com.allow_dcom</u>	"0"	PHP_INI_SYSTEM	Disponible PHP 4.0
<u>com.autoregister_casesensitive</u>	"1"	PHP_INI_ALL	PHP_INI_ALL en PHP Disponible PHP 4.7
<u>com.autoregister_typelib</u>	"0"	PHP_INI_ALL	PHP_INI_ALL pour PH Disponible PHP 4.7
<u>com.autoregister_verbose</u>	"0"	PHP_INI_ALL	PHP_INI_ALL pour PH

			Disponible PHP 4.3.8
<u>com.code_page</u>	""	PHP_INI_ALL	Disponible PHP 5.0.0
<u>com.typelib_file</u>	""	PHP_INI_SYSTEM	Disponible PHP 4.0.0
<u>crack.default_dictionary</u>	NULL	PHP_INI_PERDIR	Disponible PHP 4.0.0
<u>daffodildb.default_host</u>	"localhost"	PHP_INI_ALL	
<u>daffodildb.default_password</u>	"daffodil"	PHP_INI_ALL	
<u>daffodildb.default_socket</u>	NULL	PHP_INI_ALL	
<u>daffodildb.default_user</u>	"DAFFODIL"	PHP_INI_ALL	
<u>daffodildb.port</u>	"3456"	PHP_INI_ALL	
<u>date.default_latitude</u>	"31.7667"	PHP_INI_ALL	Disponible PHP 5.0.0
<u>date.default_longitude</u>	"35.2333"	PHP_INI_ALL	Disponible PHP 5.0.0
<u>date.sunrise_zenith</u>	"90.83"	PHP_INI_ALL	Disponible PHP 5.0.0
<u>date.sunset_zenith</u>	"90.83"	PHP_INI_ALL	Disponible PHP 5.0.0
<u>date.timezone</u>	""	PHP_INI_ALL	Disponible PHP 5.0.0
<u>dba.default_handler</u>	""	PHP_INI_ALL	Disponible PHP 4.3.8
<u>dbx.colnames_case</u>	"unchanged"	PHP_INI_SYSTEM	Disponible PHP 4.3.8
<u>default_charset</u>	""	PHP_INI_ALL	
<u>default_mimetype</u>	"text/html"	PHP_INI_ALL	
<u>default_socket_timeout</u>	"60"	PHP_INI_ALL	Disponible PHP 4.3.8
<u>define_syslog_variables</u>	"0"	PHP_INI_ALL	
<u>disable_classes</u>	""	php.ini uniquement	Disponible PHP 4.3.8
<u>disable_functions</u>	""	php.ini uniquement	Disponible PHP 4.0.0
<u>display_errors</u>	"1"	PHP_INI_ALL	
<u>display_startup_errors</u>	"0"	PHP_INI_ALL	Disponible PHP 4.0.0
<u>docref_ext</u>	""	PHP_INI_ALL	Disponible PHP 4.3.8
<u>docref_root</u>	""	PHP_INI_ALL	Disponible PHP 4.3.8
<u>doc_root</u>	NULL	PHP_INI_SYSTEM	
<u>enable_dl</u>	"1"	PHP_INI_SYSTEM	
<u>engine</u>	"1"	PHP_INI_ALL	Disponible PHP 4.0.0
<u>error_append_string</u>	NULL	PHP_INI_ALL	

<u>error_log</u>	NULL	PHP_INI_ALL	
<u>error_prepend_string</u>	NULL	PHP_INI_ALL	
<u>error_reporting</u>	NULL	PHP_INI_ALL	
<u>exif.decode_jis_intel</u>	"JIS"	PHP_INI_ALL	Disponible PHP 4.3
<u>exif.decode_jis_motorola</u>	"JIS"	PHP_INI_ALL	Disponible PHP 4.3
<u>exif.decode_unicode_intel</u>	"UCS-2LE"	PHP_INI_ALL	Disponible PHP 4.3
<u>exif.decode_unicode_motorola</u>	"UCS-2BE"	PHP_INI_ALL	Disponible PHP 4.3
<u>exif.encode_jis</u>	""	PHP_INI_ALL	Disponible PHP 4.3
<u>exif.encode_unicode</u>	"ISO-8859-15"	PHP_INI_ALL	Disponible PHP 4.3
<u>expect.logfile</u>	""	PHP_INI_ALL	
<u>expect.loguser</u>	"1"	PHP_INI_ALL	
<u>expect.timeout</u>	"10"	PHP_INI_ALL	
<u>expose_php</u>	"1"	php.ini uniquement	
<u>extension_dir</u>	"/path/to/php"	PHP_INI_SYSTEM	
<u>fastcgi.impersonate</u>	"0"	PHP_INI_ALL	
<u>fastcgi.logging</u>	"1"	PHP_INI_ALL	
<u>fbsql.allow_persistent</u>	"1"	PHP_INI_SYSTEM	Disponible PHP 4.2
<u>fbsql.autocommit</u>	"1"	PHP_INI_SYSTEM	Disponible PHP 4.0
<u>fbsql.batchsize</u>	"1000"	PHP_INI_ALL	Disponible PHP 5.3
<u>fbsql.default_database</u>	""	PHP_INI_SYSTEM	Disponible PHP 4.0
<u>fbsql.default_database_password</u>	""	PHP_INI_SYSTEM	Disponible PHP 4.0
<u>fbsql.default_host</u>	NULL	PHP_INI_SYSTEM	Disponible PHP 4.0
<u>fbsql.default_password</u>	""	PHP_INI_SYSTEM	Disponible PHP 4.0
<u>fbsql.default_user</u>	"_SYSTEM"	PHP_INI_SYSTEM	Disponible PHP 4.0
<u>fbsql.generate_warnings</u>	"0"	PHP_INI_SYSTEM	Disponible PHP 4.0
<u>fbsql.max_connections</u>	"128"	PHP_INI_SYSTEM	Disponible PHP 4.0
<u>fbsql.max_links</u>	"128"	PHP_INI_SYSTEM	Disponible PHP 4.0
<u>fbsql.max_persistent</u>	"-1"	PHP_INI_SYSTEM	Disponible PHP 4.0
<u>fbsql.max_results</u>	"128"	PHP_INI_SYSTEM	Disponible PHP 4.0

<u>file_uploads</u>	"1"	PHP_INI_SYSTEM	PHP_INI_SYSTEM depuis PHP 4.2.3. Disponible depuis PHP 4.3.8
<u>filter.default</u>	"string"	PHP_INI_ALL	
<u>filter.default_flags</u>	NULL	PHP_INI_ALL	
<u>from</u>	""	PHP_INI_ALL	
<u>highlight.bg</u>	"#FFFFFF"	PHP_INI_ALL	
<u>highlight.comment</u>	"#FF8000"	PHP_INI_ALL	
<u>highlight.default</u>	"#0000BB"	PHP_INI_ALL	
<u>highlight.html</u>	"#000000"	PHP_INI_ALL	
<u>highlight.keyword</u>	"#007700"	PHP_INI_ALL	
<u>highlight.string</u>	"#DD0000"	PHP_INI_ALL	
<u>html_errors</u>	"1"	PHP_INI_ALL	PHP_INI_ALL depuis PHP 4.2.3. Disponible depuis PHP 4.3.8
<u>http.allowed_methods</u>	""	PHP_INI_ALL	
<u>http.allowed_methods_log</u>	""	PHP_INI_ALL	
<u>http.cache_log</u>	""	PHP_INI_ALL	
<u>http.composite_log</u>	""	PHP_INI_ALL	
<u>http.etag_mode</u>	"-2"	PHP_INI_ALL	
<u>http.force_exit</u>	"1"	PHP_INI_ALL	
<u>http.only_exceptions</u>	"0"	PHP_INI_ALL	
<u>http.redirect_log</u>	""	PHP_INI_ALL	
<u>hyperwave.allow_persistent</u>	"0"	PHP_INI_SYSTEM	Disponible depuis PHP 4.3.8
<u>hyperwave.default_port</u>	"418"	PHP_INI_ALL	
<u>ibase.allow_persistent</u>	"1"	PHP_INI_SYSTEM	
<u>ibase.dateformat</u>	"%Y-%m-%d"	PHP_INI_ALL	
<u>ibase.default_charset</u>	NULL	PHP_INI_ALL	Disponible depuis PHP 5.0.0
<u>ibase.default_db</u>	NULL	PHP_INI_SYSTEM	Disponible depuis PHP 5.0.0
<u>ibase.default_password</u>	NULL	PHP_INI_ALL	
<u>ibase.default_user</u>	NULL	PHP_INI_ALL	
<u>ibase.max_links</u>	"-1"	PHP_INI_SYSTEM	
<u>ibase.max_persistent</u>	"-1"	PHP_INI_SYSTEM	
<u>ibase.timeformat</u>	"%H:%M:%S"	PHP_INI_ALL	
<u>ibase.timestampformat</u>	"%Y-%m-%d %H:%M:%S"	PHP_INI_ALL	
<u>ibm_db2.binmode</u>	"1"	PHP_INI_ALL	
<u>ibm_db2.instance_name</u>	NULL	PHP_INI_SYSTEM	
<u>iconv.input_encoding</u>	"ISO-8859-1"	PHP_INI_ALL	Disponible depuis PHP 4.0.0
<u>iconv.internal_encoding</u>	"ISO-8859-1"	PHP_INI_ALL	Disponible depuis PHP 4.0.0
<u>iconv.output_encoding</u>	"ISO-8859-1"	PHP_INI_ALL	

			Disponi PHP 4.0
<u>ifx.allow_persistent</u>	"1"	PHP_INI_SYSTEM	
<u>ifx.blobinfile</u>	"1"	PHP_INI_ALL	
<u>ifx.byteasvarchar</u>	"0"	PHP_INI_ALL	
<u>ifx.charasvarchar</u>	"0"	PHP_INI_ALL	
<u>ifx.default_host</u>	NULL	PHP_INI_SYSTEM	
<u>ifx.default_password</u>	NULL	PHP_INI_SYSTEM	
<u>ifx.default_user</u>	NULL	PHP_INI_SYSTEM	
<u>ifx.max_links</u>	"-1"	PHP_INI_SYSTEM	
<u>ifx.max_persistent</u>	"-1"	PHP_INI_SYSTEM	
<u>ifx.nullformat</u>	"0"	PHP_INI_ALL	
<u>ifx.textasvarchar</u>	"0"	PHP_INI_ALL	
<u>ignore_repeated_errors</u>	"0"	PHP_INI_ALL	Disponi PHP 4.3
<u>ignore_repeated_source</u>	"0"	PHP_INI_ALL	Disponi PHP 4.3
<u>ignore_user_abort</u>	"0"	PHP_INI_ALL	
<u>implicit_flush</u>	"0"	PHP_INI_ALL	PHP_INI depuis 4.2.3.
<u>include_path</u>	".:/path/to/php/pear"	PHP_INI_ALL	
<u>ingres.allow_persistent</u>	"1"	PHP_INI_SYSTEM	Disponi PHP 4.0
<u>ingres.blob_segment_length</u>	"4096"	PHP_INI_ALL	
<u>ingres.cursor_mode</u>	"0"	PHP_INI_ALL	
<u>ingres.default_database</u>	NULL	PHP_INI_ALL	Disponi PHP 4.0
<u>ingres.default_password</u>	NULL	PHP_INI_ALL	Disponi PHP 4.0
<u>ingres.default_user</u>	NULL	PHP_INI_ALL	Disponi PHP 4.0
<u>ingres.max_links</u>	"-1"	PHP_INI_SYSTEM	Disponi PHP 4.0
<u>ingres.max_persistent</u>	"-1"	PHP_INI_SYSTEM	Disponi PHP 4.0
<u>ingres.report_db_warnings</u>	"1"	PHP_INI_ALL	
<u>ircg.control_user</u>	"nobody"	PHP_INI_ALL	Disponi PHP 5.0
<u>ircg.keep_alive_interval</u>	"60"	PHP_INI_ALL	Disponi PHP 5.0
<u>ircg.max_format_message_sets</u>	"12"	PHP_INI_ALL	Disponi PHP 5.0
<u>ircg.shared_mem_size</u>	"6000000"	PHP_INI_ALL	Disponi PHP 5.0
<u>ircg.work_dir</u>	"/tmp/ircg"	PHP_INI_ALL	Disponi PHP 5.0
<u>last_modified</u>	"0"	PHP_INI_ALL	

			Disponible PHP 4.0
<u>ldap.max_links</u>	"-1"	PHP_INI_SYSTEM	
<u>log_errors</u>	"0"	PHP_INI_ALL	
<u>log_errors_max_len</u>	"1024"	PHP_INI_ALL	Disponible PHP 4.3
<u>magic_quotes_gpc</u>	"1"	PHP_INI_PERDIR	PHP_INI_ALL depuis PHP 4.2.3.
<u>magic_quotes_runtime</u>	"0"	PHP_INI_ALL	
<u>magic_quotes_sybase</u>	"0"	PHP_INI_ALL	
<u>mail.force_extra_parameters</u>	NULL	PHP_INI_PERDIR	Disponible PHP 5.0
<u>mailparse.def_charset</u>	"us-ascii"	PHP_INI_ALL	Disponible PHP 4.7
<u>maxdb.default_db</u>	NULL	PHP_INI_ALL	
<u>maxdb.default_host</u>	NULL	PHP_INI_ALL	
<u>maxdb.default_pw</u>	NULL	PHP_INI_ALL	
<u>maxdb.default_user</u>	NULL	PHP_INI_ALL	
<u>maxdb.long_readlen</u>	"200"	PHP_INI_ALL	
<u>max_execution_time</u>	"30"	PHP_INI_ALL	
<u>max_input_time</u>	"-1"	PHP_INI_PERDIR	Disponible PHP 4.3
<u>mbstring.detect_order</u>	NULL	PHP_INI_ALL	Disponible PHP 4.0
<u>mbstring.encoding_translation</u>	"0"	PHP_INI_PERDIR	Disponible PHP 4.3
<u>mbstring.func_overload</u>	"0"	PHP_INI_PERDIR	PHP_INI_ALL depuis PHP 4.2.3. D depuis PHP
<u>mbstring.http_input</u>	"pass"	PHP_INI_ALL	Disponible PHP 4.0
<u>mbstring.http_output</u>	"pass"	PHP_INI_ALL	Disponible PHP 4.0
<u>mbstring.internal_encoding</u>	NULL	PHP_INI_ALL	Disponible PHP 4.0
<u>mbstring.language</u>	"neutral"	PHP_INI_PERDIR	Disponible PHP 4.3
<u>mbstring.script_encoding</u>	NULL	PHP_INI_ALL	Disponible PHP 4.3
<u>mbstring.substitute_character</u>	NULL	PHP_INI_ALL	Disponible PHP 4.0
<u>mcrypt.algorithms_dir</u>	NULL	PHP_INI_ALL	Disponible PHP 4.0
<u>mcrypt.modes_dir</u>	NULL	PHP_INI_ALL	Disponible PHP 4.0
<u>memory_limit</u>	"8M"	PHP_INI_ALL	

<u>mime_magic.debug</u>	"0"	PHP_INI_SYSTEM	Disponible PHP 5.0
<u>mime_magic.magicfile</u>	"/path/to/php/magic.mime"	PHP_INI_SYSTEM	Disponible PHP 4.3
<u>mssql.allow_persistent</u>	"1"	PHP_INI_ALL	
<u>mssql.max_links</u>	"-1"	PHP_INI_ALL	
<u>mssql.max_persistent</u>	"-1"	PHP_INI_ALL	
<u>mssql.allow_persistent</u>	"1"	PHP_INI_SYSTEM	
<u>mssql.batchsize</u>	"0"	PHP_INI_ALL	Disponible PHP 4.0
<u>mssql.compatability_mode</u>	"0"	PHP_INI_ALL	
<u>mssql.connect_timeout</u>	"5"	PHP_INI_ALL	
<u>mssql.datetimeconvert</u>	"1"	PHP_INI_ALL	Disponible PHP 4.2
<u>mssql.max_links</u>	"-1"	PHP_INI_SYSTEM	
<u>mssql.max_persistent</u>	"-1"	PHP_INI_SYSTEM	
<u>mssql.max_procs</u>	"-1"	PHP_INI_ALL	Disponible PHP 4.3
<u>mssql.min_error_severity</u>	"10"	PHP_INI_ALL	
<u>mssql.min_message_severity</u>	"10"	PHP_INI_ALL	
<u>mssql.secure_connection</u>	"0"	PHP_INI_SYSTEM	Disponible PHP 4.3
<u>mssql.textlimit</u>	"-1"	PHP_INI_ALL	
<u>mssql.textsize</u>	"-1"	PHP_INI_ALL	
<u>mssql.timeout</u>	"60"	PHP_INI_ALL	Disponible PHP 4.3
<u>mysql.allow_persistent</u>	"1"	PHP_INI_SYSTEM	
<u>mysql.connect_timeout</u>	"60"	PHP_INI_ALL	PHP_INI_ALL depuis PHP 4.3.2. Disponible depuis PHP 4.3
<u>mysql.default_host</u>	NULL	PHP_INI_ALL	
<u>mysql.default_password</u>	NULL	PHP_INI_ALL	
<u>mysql.default_port</u>	NULL	PHP_INI_ALL	
<u>mysql.default_socket</u>	NULL	PHP_INI_ALL	Disponible PHP 4.0
<u>mysql.default_user</u>	NULL	PHP_INI_ALL	
<u>mysql.max_links</u>	"-1"	PHP_INI_SYSTEM	
<u>mysql.max_persistent</u>	"-1"	PHP_INI_SYSTEM	
<u>mysql.trace_mode</u>	"0"	PHP_INI_ALL	Disponible PHP 4.3
<u>mysqli.default_host</u>	NULL	PHP_INI_ALL	Disponible PHP 5.0
<u>mysqli.default_port</u>	"3306"	PHP_INI_ALL	Disponible PHP 5.0
<u>mysqli.default_pw</u>	NULL	PHP_INI_ALL	Disponible PHP 5.0

<u>mysqli.default_socket</u>	NULL	PHP_INI_ALL	Disponible PHP 5.0
<u>mysqli.default_user</u>	NULL	PHP_INI_ALL	Disponible PHP 5.0
<u>mysqli.max_links</u>	"-1"	PHP_INI_SYSTEM	Disponible PHP 5.0
<u>mysqli.reconnect</u>	"0"	PHP_INI_SYSTEM	Disponible PHP 5.0
<u>namazu.debugmode</u>	"0"	PHP_INI_ALL	
<u>namazu.lang</u>	NULL	PHP_INI_ALL	
<u>namazu.loggingmode</u>	"0"	PHP_INI_ALL	
<u>namazu.sortmethod</u>	NULL	PHP_INI_ALL	
<u>namazu.sortorder</u>	NULL	PHP_INI_ALL	
<u>nsapi.read_timeout</u>	"60"	PHP_INI_ALL	Disponible PHP 4.3
<u>oci8.default_prefetch</u>	"10"	PHP_INI_SYSTEM	
<u>oci8.max_persistent</u>	"-1"	PHP_INI_SYSTEM	
<u>oci8.old_oci_close_semantics</u>	"0"	PHP_INI_SYSTEM	
<u>oci8.persistent_timeout</u>	"-1"	PHP_INI_SYSTEM	
<u>oci8.ping_interval</u>	"60"	PHP_INI_SYSTEM	
<u>oci8.privileged_connect</u>	"0"	PHP_INI_SYSTEM	
<u>oci8.statement_cache_size</u>	"20"	PHP_INI_SYSTEM	
<u>odbc.allow_persistent</u>	"1"	PHP_INI_SYSTEM	
<u>odbc.check_persistent</u>	"1"	PHP_INI_SYSTEM	
<u>odbc.defaultbinmode</u>	"1"	PHP_INI_ALL	
<u>odbc.defaultlrl</u>	"4096"	PHP_INI_ALL	
<u>odbc.default_db</u>	NULL	PHP_INI_ALL	
<u>odbc.default_pw</u>	NULL	PHP_INI_ALL	
<u>odbc.default_user</u>	NULL	PHP_INI_ALL	
<u>odbc.max_links</u>	"-1"	PHP_INI_SYSTEM	
<u>odbc.max_persistent</u>	"-1"	PHP_INI_SYSTEM	
<u>opendirectory.max_refs</u>	"-1"	PHP_INI_ALL	
<u>opendirectory.separator</u>	"/"	PHP_INI_ALL	
<u>open_basedir</u>	NULL	PHP_INI_SYSTEM	
<u>oracle.allow_persistent</u>	"-1"	PHP_INI_ALL	
<u>oracle.max_links</u>	"-1"	PHP_INI_ALL	
<u>oracle.max_persistent</u>	"-1"	PHP_INI_ALL	
<u>output_buffering</u>	"0"	PHP_INI_PERDIR	
<u>output_handler</u>	NULL	PHP_INI_PERDIR	Disponible PHP 4.0
<u>pdo_odbc.connection_pooling</u>	"strict"	PHP_INI_ALL	
<u>pfpro.defaulthost</u>	"test-payflow.verisign.com"	PHP_INI_ALL	Disponible PHP 4.0
<u>pfpro.defaultport</u>	"443"	PHP_INI_ALL	Disponible PHP 4.0
<u>pfpro.defaulttimeout</u>	"30"	PHP_INI_ALL	

			Disponi PHP 4.0
<u>pfpro.proxyaddress</u>	""	PHP_INI_ALL	Disponi PHP 4.0
<u>pfpro.proxylogon</u>	""	PHP_INI_ALL	Disponi PHP 4.0
<u>pfpro.proxypassword</u>	""	PHP_INI_ALL	Disponi PHP 4.0
<u>pfpro.proxyport</u>	""	PHP_INI_ALL	Disponi PHP 4.0
<u>pgsql.allow_persistent</u>	"1"	PHP_INI_SYSTEM	
<u>pgsql.auto_reset_persistent</u>	"0"	PHP_INI_SYSTEM	Disponi PHP 4.2
<u>pgsql.ignore_notice</u>	"0"	PHP_INI_ALL	Disponi PHP 4.3
<u>pgsql.log_notice</u>	"0"	PHP_INI_ALL	Disponi PHP 4.3
<u>pgsql.max_links</u>	"-1"	PHP_INI_SYSTEM	
<u>pgsql.max_persistent</u>	"-1"	PHP_INI_SYSTEM	
<u>post_max_size</u>	"8M"	PHP_INI_PERDIR	PHP_INI depuis l 4.2.3. D depuis l
<u>precision</u>	"14"	PHP_INI_ALL	
<u>printer.default_printer</u>	""	PHP_INI_ALL	
<u>realpath_cache_size</u>	"16K"	PHP_INI_SYSTEM	Disponi PHP 5.0
<u>realpath_cache_ttl</u>	"120"	PHP_INI_SYSTEM	Disponi PHP 5.0
<u>register_argc_argv</u>	"1"	PHP_INI_PERDIR	PHP_INI depuis l 4.2.3.
<u>register_globals</u>	"0"	PHP_INI_PERDIR	PHP_INI depuis l 4.2.3.
<u>register_long_arrays</u>	"1"	PHP_INI_PERDIR	Disponi PHP 5.0
<u>report_memleaks</u>	"1"	PHP_INI_ALL	Disponi PHP 4.3
<u>report zend debug</u>	"1"	PHP_INI_ALL	Disponi PHP 5.0
<u>runkit.internal_override</u>	"0"	PHP_INI_SYSTEM	
<u>runkit.superglobal</u>	""	PHP_INI_PERDIR	
<u>safe_mode</u>	"0"	PHP_INI_SYSTEM	
<u>safe_mode_allowed_env_vars</u>	"PHP_"	PHP_INI_SYSTEM	
<u>safe_mode_exec_dir</u>	""	PHP_INI_SYSTEM	
<u>safe_mode_gid</u>	"0"	PHP_INI_SYSTEM	Disponi PHP 4.1

<u>safe_mode_include_dir</u>	NULL	PHP_INI_SYSTEM	Disponible PHP 4.3.0
<u>safe_mode_protected_env_vars</u>	"LD_LIBRARY_PATH"	PHP_INI_SYSTEM	
<u>sendmail_from</u>	NULL	PHP_INI_ALL	
<u>sendmail_path</u>	NULL	PHP_INI_SYSTEM	
<u>serialize_precision</u>	"100"	PHP_INI_ALL	Disponible PHP 4.3.0
<u>session.auto_start</u>	"0"	PHP_INI_ALL	
<u>session.bug_compat_42</u>	"1"	PHP_INI_ALL	Disponible PHP 4.3.0
<u>session.bug_compat_warn</u>	"1"	PHP_INI_ALL	Disponible PHP 4.3.0
<u>session.cache_expire</u>	"180"	PHP_INI_ALL	
<u>session.cache_limiter</u>	"nocache"	PHP_INI_ALL	
<u>session.cookie_domain</u>	""	PHP_INI_ALL	
<u>session.cookie_lifetime</u>	"0"	PHP_INI_ALL	
<u>session.cookie_path</u>	"/"	PHP_INI_ALL	
<u>session.cookie_secure</u>	""	PHP_INI_ALL	Disponible PHP 4.0.0
<u>session.entropy_file</u>	""	PHP_INI_ALL	
<u>session.entropy_length</u>	"0"	PHP_INI_ALL	
<u>session.gc_divisor</u>	"100"	PHP_INI_ALL	Disponible PHP 4.3.0
<u>session.gc_maxlifetime</u>	"1440"	PHP_INI_ALL	
<u>session.gc_probability</u>	"1"	PHP_INI_ALL	
<u>session.hash_bits_per_character</u>	"4"	PHP_INI_ALL	Disponible PHP 5.0.0
<u>session.hash_function</u>	"0"	PHP_INI_ALL	Disponible PHP 5.0.0
<u>session.name</u>	"PHPSESSID"	PHP_INI_ALL	
<u>session.referer_check</u>	""	PHP_INI_ALL	
<u>session.save_handler</u>	"files"	PHP_INI_ALL	
<u>session.save_path</u>	""	PHP_INI_ALL	
<u>session.serialize_handler</u>	"php"	PHP_INI_ALL	
<u>session.use_cookies</u>	"1"	PHP_INI_ALL	
<u>session.use_only_cookies</u>	"0"	PHP_INI_ALL	Disponible PHP 4.3.0
<u>session.use_trans_sid</u>	"0"	PHP_INI_ALL	PHP_INI_ALL depuis PHP 4.2.3. PHP_INI_ALL depuis PHP 4.0.0 Disponible PHP 4.0.0
<u>session_pgsql.create_table</u>	"1"	PHP_INI_SYSTEM	
<u>session_pgsql.db</u>	"host=localhost dbname=php_session user=nobody"	PHP_INI_SYSTEM	
<u>session_pgsql.disable</u>	"0"	PHP_INI_SYSTEM	

<u>session_pgsql.failover_mode</u>	"0"	PHP_INI_SYSTEM	
<u>session_pgsql.gc_interval</u>	"3600"	PHP_INI_SYSTEM	
<u>session_pgsql.keep_expired</u>	"0"	PHP_INI_SYSTEM	
<u>session_pgsql.sem_file_name</u>	"/tmp/php_session_pgsql"	PHP_INI_SYSTEM	
<u>session_pgsql.serializeable</u>	"0"	PHP_INI_SYSTEM	
<u>session_pgsql.short_circuit</u>	"0"	PHP_INI_SYSTEM	
<u>session_pgsql.use_app_vars</u>	"0"	PHP_INI_SYSTEM	
<u>session_pgsql.vacuum_interval</u>	"21600"	PHP_INI_SYSTEM	
<u>short_open_tag</u>	"1"	PHP_INI_PERDIR	PHP_INI_PERDIR depuis PHP 4.0.0.
<u>simple_cvs.authMethod</u>	"0"	PHP_INI_ALL	
<u>simple_cvs.compressionLevel</u>	"0"	PHP_INI_ALL	
<u>simple_cvs.cvsRoot</u>	"0"	PHP_INI_ALL	
<u>simple_cvs.host</u>	"0"	PHP_INI_ALL	
<u>simple_cvs.moduleName</u>	"0"	PHP_INI_ALL	
<u>simple_cvs.userName</u>	"0"	PHP_INI_ALL	
<u>simple_cvs.workingDir</u>	"0"	PHP_INI_ALL	
<u>SMTP</u>	"localhost"	PHP_INI_ALL	
<u>smtp_port</u>	"25"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
<u>soap.wsdl_cache_dir</u>	"/tmp"	PHP_INI_ALL	Disponible depuis PHP 5.0.0.
<u>soap.wsdl_cache_enabled</u>	"1"	PHP_INI_ALL	Disponible depuis PHP 5.0.0.
<u>soap.wsdl_cache_ttl</u>	"86400"	PHP_INI_ALL	Disponible depuis PHP 5.0.0.
<u>sql.safe_mode</u>	"0"	PHP_INI_SYSTEM	
<u>sqlite.assoc_case</u>	"0"	PHP_INI_ALL	Disponible depuis PHP 5.0.0.
<u>sybase.allow_persistent</u>	"1"	PHP_INI_ALL	Disponible depuis PHP 4.0.0.
<u>sybase.compatability_mode</u>	"0"	PHP_INI_ALL	
<u>sybase.interface_file</u>	""	PHP_INI_ALL	
<u>sybase.max_links</u>	"-1"	PHP_INI_ALL	Disponible depuis PHP 4.0.0.
<u>sybase.max_persistent</u>	"-1"	PHP_INI_ALL	Disponible depuis PHP 4.0.0.
<u>sybase.min_error_severity</u>	"10"	PHP_INI_ALL	
<u>sybase.min_message_severity</u>	"10"	PHP_INI_ALL	
<u>sybct.allow_persistent</u>	"1"	PHP_INI_SYSTEM	Disponible depuis PHP 4.0.0.
<u>sybct.deadlock_retry_count</u>	"0"	PHP_INI_ALL	Disponible depuis PHP 4.3.0.
<u>sybct.hostname</u>	NULL	PHP_INI_ALL	Disponible depuis PHP 4.0.0.
<u>sybct.login_timeout</u>	"-1"	PHP_INI_ALL	

			Disponible PHP 4.3.
<u>sybct.max_links</u>	"-1"	PHP_INI_SYSTEM	Disponible PHP 4.0.
<u>sybct.max_persistent</u>	"-1"	PHP_INI_SYSTEM	Disponible PHP 4.0.
<u>sybct.min_client_severity</u>	"10"	PHP_INI_ALL	Disponible PHP 4.0.
<u>sybct.min_server_severity</u>	"10"	PHP_INI_ALL	Disponible PHP 4.0.
<u>sybct.packet_size</u>	"0"	PHP_INI_ALL	
<u>sybct.timeout</u>	"0"	PHP_INI_ALL	
<u>sysvshm.init_mem</u>	"10000"	PHP_INI_ALL	
<u>tidy.clean_output</u>	"0"	PHP_INI_PERDIR	Disponible PHP 5.0.
<u>tidy.default_config</u>	""	PHP_INI_SYSTEM	Disponible PHP 5.0.
<u>track_errors</u>	"0"	PHP_INI_ALL	
<u>unicode.fallback_encoding</u>	NULL	PHP_INI_ALL	
<u>unicode.from_error_mode</u>	"2"	PHP_INI_ALL	
<u>unicode.from_error_subst_char</u>	"3f"	PHP_INI_ALL	
<u>unicode.http_input_encoding</u>	NULL	PHP_INI_ALL	
<u>unicode.output_encoding</u>	NULL	PHP_INI_ALL	
<u>unicode.runtime_encoding</u>	NULL	PHP_INI_ALL	
<u>unicode.script_encoding</u>	NULL	PHP_INI_ALL	
<u>unicode_semantics</u>	"off"	PHP_INI_PERDIR	
<u>unserialize_callback_func</u>	NULL	PHP_INI_ALL	Disponible PHP 4.2.
<u>upload_max_filesize</u>	"2M"	PHP_INI_PERDIR	PHP_INI_PERDIR depuis la version 4.2.3.
<u>upload_tmp_dir</u>	NULL	PHP_INI_SYSTEM	
<u>url_rewriter.tags</u>	"a=href,area=href,frame=src,form=,fieldset="	PHP_INI_ALL	Disponible PHP 4.0.
<u>user_agent</u>	NULL	PHP_INI_ALL	Disponible PHP 4.3.
<u>user_dir</u>	NULL	PHP_INI_SYSTEM	
<u>valkyrie.auto_validate</u>	"0"	PHP_INI_ALL	
<u>valkyrie.config_path</u>	NULL	PHP_INI_ALL	
<u>variables_order</u>	"EGPCS"	PHP_INI_PERDIR	
<u>xbithack</u>	"0"	PHP_INI_ALL	Disponible PHP 4.0.
<u>xmlrpc_errors</u>	"0"	PHP_INI_SYSTEM	Disponible PHP 4.1.
<u>xmlrpc_error_number</u>	"0"	PHP_INI_ALL	Disponible PHP 4.1.
<u>xmms.path</u>	"/usr/bin/xmms"	PHP_INI_ALL	

xmms.session	"0"	PHP_INI_ALL	
y2k_compliance	"1"	PHP_INI_ALL	
yaz.keepalive	"120"	PHP_INI_ALL	
yaz.log_file	NULL	PHP_INI_ALL	Disponible PHP 4.3
yaz.max_links	"100"	PHP_INI_ALL	Disponible PHP 4.3
zend.ze1_compatibility_mode	"0"	PHP_INI_ALL	Disponible PHP 5.0
zlib.output_compression	"0"	PHP_INI_ALL	Disponible PHP 4.0
zlib.output_compression_level	"-1"	PHP_INI_ALL	Disponible PHP 4.3
zlib.output_handler	""	PHP_INI_ALL	Disponible PHP 4.3

Constante	Valeur	Signification
PHP_INI_USER	1	Les entrées peuvent être définies dans les scripts utilisateurs ou dans le <u>registre de Windows</u>
PHP_INI_PERDIR	2	Les entrées peuvent être définies dans le php.ini , un .htaccess ou le httpd.conf
PHP_INI_SYSTEM	4	Les entrées peuvent être définies dans le php.ini ou le httpd.conf
PHP_INI_ALL	7	Les entrées peuvent être définies n'importe où.

11.7.2 Description des directives internes du php.ini

Cette liste inclut les directives internes du php.ini que vous pouvez définir pour personnaliser votre configuration de PHP. Les directives gérées par les extensions sont listées et détaillées dans les pages de documentation respectives des extensions ; Les informations concernant les directives sur les sessions par exemple, peuvent être trouvées sur la page de documentation des sessions .

11.7.2.1 Options Httpd

Nom	Par défaut	Modifiable	Historique
async_send	"0"	PHP_INI_ALL	

11.7.2.2 Options du langage

Nom	Par défaut	Modifiable	Historique
short_open_tag	On	PHP_INI_SYSTEM PHP_INI_PERDIR	PHP_INI_ALL en PHP <= 4.0.0.
asp_tags	"0"	PHP_INI_PERDIR	PHP_INI_ALL en PHP <= 4.0.0.
precision	"14"	PHP_INI_ALL	
y2k_compliance	"1"	PHP_INI_ALL	
allow_call_time_pass_reference	"1"	PHP_INI_PERDIR	PHP_INI_ALL en

			PHP <= 4.0.0.
expose_php	"1"	php.ini uniquement	
zend.ze1_compatibility_mode	"0"	PHP_INI_ALL	Disponible depuis PHP 5.0.0.

Voici un éclaircissement sur l'utilisation des directives de configuration.

short_open_tag boolean

Définit si les balises courtes d'ouverture de PHP (`<? ?>`) sont autorisées ou non. Si vous voulez utiliser PHP avec XML, vous devez désactiver cette option de configuration pour pouvoir utiliser `<?xml ?>` . Sinon, vous pouvez l'écrire à l'aide de PHP, par exemple : `<?php echo '<?xml version="1.0"; ?>'` . Si cette option est désactivée, vous devez utiliser la version longue d'ouverture de balises PHP (`<?php ?>`).

Note

Cette directive affecte également l'utilisation de `<?=` , qui est identique à `<? echo` . L'utilisation de cette écriture nécessite que l'option `short_open_tag` soit activée. to be on.

asp_tags boolean

Active l'utilisation des balises ASP (`<% %>`) tout en conservant les balises PHP (`<?php ?>`). Cela inclut l'utilisation des balises courtes comme `<%= $valeur %>`. Pour plus d'informations, lisez la page [Sortir du mode HTML](#) .

Note

Le support des balises ASP a été ajoutée en PHP 3.0.4.

precision integer

Le nombre de décimal significatif à afficher dans les nombres à virgule flottante.

y2k_compliance boolean

Force la conformité avec l'an 2000 (peut être la cause de problèmes avec les navigateurs non-conformes)

allow_call_time_pass_reference boolean

Active ou non la possibilité de forcer les arguments à être passés par référence lors de l'appel à une fonction. Cette méthode est dépréciée et sera très certainement plus supportée dans les futures versions de PHP/Zend. Il est préférable de spécifier directement dans la déclaration de la fonction si les arguments seront passés ou non par référence. Nous vous encourageons à désactiver cette option et de vous assurer que vos scripts fonctionnent correctement avec dans le but de vous assurez qu'ils fonctionneront avec les versions futures du langage (vous devriez recevoir une alerte à chaque fois que vous utiliserez cette fonctionnalité et que l'argument passé par valeur au lieu de l'être par référence).

Passer les arguments par référence à l'appel de la fonction a été abandonné pour rendre le code plus clair. Les fonctions risquaient de modifier les arguments d'une manière aléatoire si l'argument était passé par référence. Pour éviter cette effet, il est préférable de spécifier le passage par référence dans le prototype de la fonction.

Voir aussi [l'explication sur les références](#) .

expose_php boolean

Décide ou non si PHP doit être affiché comme étant installé sur le serveur (e.g. en ajoutant sa signature dans les en-têtes du serveur web). Ce n'est en aucun cas une menace de sécurité, mais cela permet de déterminer si vous utilisez PHP ou non sur votre serveur.

zend.ze1_compatibility_mode boolean

Active le mode de compatibilité avec le Zend Engine 1 (PHP 4). Cela affecte le clonage, le casting et la [comparaison des objets](#) . Dans ce mode, les objets sont

passés par valeur plutôt que par référence, par défaut.

11.7.2.3 Limite des ressources

Nom	Par défaut	Modifiable	Historique
memory_limit	"8M"	PHP_INI_ALL	

Voici un éclaircissement sur l'utilisation des directives de configuration.

memory_limit integer

Cette option détermine la mémoire limite, en octets, qu'un script est autorisé à allouer. Cela permet de prévenir l'utilisation de toute la mémoire par un script mal codé. Pour utiliser cette directive, vous devez l'avoir activé lors de la compilation. Du coup, votre ligne de configuration devait ressembler à cela : --enable-memory-limit . Notez que vous devez positionner cette directive à -1 si vous ne voulez aucune limite.

Depuis PHP 4.3.2 et lorsque memory_limit est actif, la fonction PHP `memory_get_usage` est disponible.

Lorsqu'un entier est utilisé, sa valeur est mesuré en octets. Vous pouvez également utiliser la notation sténographique comme décrit dans cette [entrée de la FAQ](#) .

Voir aussi : [max_execution_time](#) .

11.7.2.4 Gestion des données

Nom	Par défaut	Modifiable	Historique
track_vars	"On"	PHP_INI_??	
arg_separator.output	"&"	PHP_INI_ALL	Disponible depuis PHP 4.0.5.
arg_separator.input	"&"	PHP_INI_PERDIR	Disponible depuis PHP 4.0.5.
variables_order	"EGPCS"	PHP_INI_PERDIR	
auto_globals_jit	"1"	PHP_INI_PERDIR	Disponible depuis PHP 5.0.0.
register_globals	"0"	PHP_INI_PERDIR	PHP_INI_ALL en PHP <= 4.2.3.
register_argc_argv	"1"	PHP_INI_PERDIR	PHP_INI_ALL en PHP <= 4.2.3.
register_long_arrays	"1"	PHP_INI_PERDIR	Disponible depuis PHP 5.0.0.
post_max_size	"8M"	PHP_INI_PERDIR	PHP_INI_SYSTEM en PHP <= 4.2.3. Disponible depuis PHP 4.0.3.
gpc_order	"GPC"	PHP_INI_ALL	
auto_prepend_file	NULL	PHP_INI_PERDIR	PHP_INI_ALL en PHP <= 4.2.3.
auto_append_file	NULL	PHP_INI_PERDIR	PHP_INI_ALL en PHP <= 4.2.3.
default_mimetype	"text/html"	PHP_INI_ALL	
default_charset	""	PHP_INI_ALL	
always_populate_raw_post_data	"0"	PHP_INI_PERDIR	PHP_INI_ALL en PHP <= 4.2.3. Disponible depuis PHP 4.1.0.
allow_webdav_methods	"0"	PHP_INI_PERDIR	

Voici un éclaircissement sur l'utilisation des directives de configuration.

`track_vars` boolean

Si cette option est configurée, alors les variables d'environnement GET, POST, Cookie et Server peuvent être trouvées respectivement dans les tableaux associatifs globaux `$_ENV`, `$_GET`, `$_POST`, `$_COOKIE` et `$_SERVER`.

Notez que depuis PHP 4.0.3, `track_vars` est toujours activée.

`arg_separator.output` string

Le séparateur utilisé lorsque PHP génère les URLs pour séparer les arguments.

`arg_separator.input` string

Liste des séparateur(s) utilisé(s) par PHP pour analyser les URLs entrantes et en déduire les valeurs.

Note

Chaque caractère de cette directive est considéré comme étant un séparateur !

`variables_order` string

Définit l'ordre de parcours des variables EGPCS (Environment, GET, POST, Cookie, Server). La valeur par défaut de cette directive est "EGPCS". Si vous définissez cette directive à "GP" par exemple, cela fera que PHP ignorera complètement les variables d'environnement, cookies et serveur et effacera toutes les variables passées en GET avec les variables de même noms passées en méthode POST.

Voir aussi [register_globals](#).

`auto_globals_jit` boolean

Lorsque cette directive est activée, les variables SERVER et ENV sont créées lorsqu'elles sont utilisées : juste lorsque c'est nécessaire. Si ces variables ne sont pas utilisées dans un script, le script verra un gain de performances.

Les directives PHP [register_globals](#), [register_long_arrays](#), et [register_argc_argv](#) doivent être désactivées pour que cette directive soit utilisable.

`register_globals` boolean

Définit si oui ou non les variables EGPCS (Environment, GET, POST, Cookie, Server) seront enregistrées comme des variables globales.

Depuis [PHP 4.2.0](#), la valeur par défaut de cette directive est **off**.

Lisez le chapitre sur la sécurité concernant [l'utilisation de register_globals](#) pour plus d'informations.

Notez que `register_globals` ne peut pas être défini durant le traitement (`ini_set`). Vous pouvez utiliser `.htaccess` si votre hôte vous le permet comme décrit ci-dessous. Exemple d'une entrée `.htaccess` : `php_flag register_globals off`.

Note

La directive <code>register_globals</code> est affectée par la directive variables_order .
--

`register_argc_argv` boolean

Dit à PHP s'il doit déclarer ou pas les variables `argv` et `argc` (qui contiendront les informations GET).

Voir aussi les [lignes de commande](#). Cette directive a été introduite depuis PHP 4.0.0 et valait toujours "on" avant.

`register_long_arrays` boolean

Dit à PHP si oui ou non il doit enregistrer les types dépréciés `$HTTP_*_VARS` comme [variables pré-définies](#). Lorsque cette directive est activé (par défaut), les variables longues PHP comme `$HTTP_GET_VARS` seront définies. Si vous ne les utilisez pas, il est recommandé de désactiver cette option pour des raisons de

performance. A la place, utilisez les variables tableaux super-globales comme `$_GET` .

Cette directive est disponible depuis PHP 5.0.0.

`post_max_size` integer

Définit la taille maximale des données reçues par méthode POST. Cette option affecte également les fichiers chargés. Pour charger de gros fichiers, cette valeur doit être plus grande que la valeur de `upload_max_filesize` .

Si la limitation de mémoire est activée par votre script de configuration, `memory_limit` affectera également les fichiers chargés. De façon général, `memory_limit` doit être plus grand que `post_max_size` .

Lorsqu'un entier est utilisé, sa valeur est mesuré en octets. Vous pouvez également utiliser la notation sténographique comme décrit dans cette [entrée de la FAQ](#) .

Dans le cas où la taille des données reçues par la méthode POST est plus grande que `post_max_size` , les [superglobales](#) `$_POST` et `$_FILES` seront vides. Ceci peut être surveillé de différentes façons, e.g. en passant une variable `$_GET` au script qui traite les données, i.e. `<form action="edit.php?processed=1">` , et ainsi vérifier si `$_GET['processed']` est défini.

`gpc_order` string

Définit l'ordre de parcours des variables GET/POST/COOKIE. Par défaut, cette directive vaut "GPC". Si vous positionnez cette directive à "GP" par exemple, PHP ignorera complètement les cookies et effacera toutes les variables de la méthode GET avec les variables de la méthode POST portant le même nom.

Note

Cette option n'est pas disponible en PHP 4. Utilisez <code>variables_order</code> à la place.

`auto_prepend_file` string

Spécifie le nom d'un fichier qui sera automatiquement parcouru avant le fichier principal. Ce fichier est inclus comme s'il l'avait été avec la fonction `include` , donc `include_path` est utilisé.

La valeur spéciale `none` désactive l'ajout automatique.

`auto_append_file` string

Spécifie le nom du fichier qui sera automatiquement parcouru après la fichier principal. Ce fichier est inclus comme s'il l'avait été avec la fonction `include` , donc `include_path` est utilisé.

La valeur spéciale `none` désactive l'ajout automatique.

Note

Si le script est terminé par la fonction <code>exit</code> , l'ajout automatique ne se fera pas .
--

`default_mimetype` string

`default_charset` string

Depuis PHP 4.0b4, PHP enverra toujours un jeu de caractères par défaut à l'en-tête HTTP Content-type: header . Pour désactiver l'envoi du jeu de caractères, définissez le tout simplement à une valeur vide.

`always_populate_raw_post_data` boolean

Remplit toujours la variable `$HTTP_RAW_POST_DATA` . Sinon, la variable est rempli uniquement des types MIME non reconnu des données. Cependant, la méthode préférée pour accès à ce type de données est `php://input` .

`$HTTP_RAW_POST_DATA` n'est pas disponible avec `enctype="multipart/form-data"`

`allow_webdav_methods` boolean

Autorise la gestion des requête HTTP WebDav avec des scripts PHP (e.g PROPFIND, PROPPATCH, MOVE, COPY, etc.). Cette directive n'existe plus depuis PHP 4.3.2. Si vous voulez récupérer les données envoyées avec la méthode POST de ce type de requêtes, vous devez également définir `always_populate_raw_post_data` .

Voir aussi : `magic_quotes_gpc` , `magic_quotes_runtime` et `magic_quotes_sybase` .

11.7.2.5 Chemins et répertoires

Nom	Par défaut	Modifiable	Historique
<code>include_path</code>	<code>"/chemin/vers/php/pear"</code>	PHP_INI_ALL	
<code>doc_root</code>	NULL	PHP_INI_SYSTEM	
<code>user_dir</code>	NULL	PHP_INI_SYSTEM	
<code>extension_dir</code>	<code>"/chemin/vers/php"</code>	PHP_INI_SYSTEM	
<code>cgi.fix_pathinfo</code>	<code>"1"</code>	PHP_INI_ALL	
<code>cgi.force_redirect</code>	<code>"1"</code>	PHP_INI_ALL	
<code>cgi.redirect_status_env</code>	NULL	PHP_INI_ALL	
<code>fastcgi.impersonate</code>	<code>"0"</code>	PHP_INI_ALL	
<code>cgi.rfc2616_headers</code>	<code>"0"</code>	PHP_INI_ALL	

Voici un éclaircissement sur l'utilisation des directives de configuration.

`include_path` string

Spécifie une liste de répertoires où les fonctions `require` , `include` et `fopen_with_path` chercheront les fichiers. Le format est identique à la variable d'environnement système **PATH** : une liste de répertoires séparés par deux points (:) sous Unix ou par un point-virgule (;) sous Windows.

<code>include_path</code> sous Unix
<code>include_path="/php/includes"</code>

<code>include_path</code> sous Windows
<code>include_path=".;c:\php\includes"</code>

L'utilisation d'un point (.) dans le chemin d'inclusion vous permet de faire des inclusions relatives au répertoire courant.

`doc_root` string

Le dossier racine de PHP sur le serveur. Uniquement utilisé si non vide. Si PHP est configuré avec le `safe_mode` , aucun fichier n'est servi en dehors de ce répertoire. Si PHP n'a pas été compilé avec `FORCE_REDIRECT`, vous **devez** définir le `doc_root` si vous utilisez PHP en tant que CGI sous n'importe quel serveur web (autre que IIS). Alternativement, vous pouvez utiliser la configuration `cgi.force_redirect` .

`user_dir` string

Le nom de base du répertoire utilisé dans un répertoire utilisateur pour les fichiers PHP, par exemple, `public_html` .

`extension_dir` string

Spécifie le répertoire dans lequel PHP doit chercher des extensions externes à charger. Voir aussi `enable_dl` et `dl` .

`extension` string

Quelles extensions doivent être chargées dynamiquement lors du démarrage de PHP.

cgi.fix_pathinfo boolean

Fournit un **réel** PATH_INFO/PATH_TRANSLATED pour CGI. Le comportement précédent de PHP était de définir PATH_TRANSLATED en SCRIPT_FILENAME et de ne pas remplir PATH_INFO. Pour plus d'informations sur PATH_INFO, lisez les spécificités CGI. Si définie à 1, PHP CGI fixera ce chemin suivant les spécifications. Si définie à 0, PHP appliquera l'ancien comportement. Par défaut, cette directive vaut 0. Vous devriez modifier vos scripts pour utiliser SCRIPT_FILENAME à la place de PATH_TRANSLATED.

cgi.force_redirect boolean

cgi.force_redirect est nécessaire pour des raisons de sécurité lors de l'utilisation de PHP en mode CGI sous la plus part des serveurs web. Si vous ne la définissez pas, PHP l'activera automatiquement par défaut. Vous pouvez la désactiver **à vos risques et périls** .

Note
Utilisateurs de Windows : Vous pouvez désactiver cette option en toute sécurité si vous utilisez le serveur web IIS ; en faite, vous le devez . Pour faire fonctionner les serveurs web OmniHTTPD ou Xitami, vous devez désactiver cette directive.

cgi.redirect_status_env string

Si cgi.force_redirect est activé et que vous ne tournez pas sous un serveur web Apache ou Netscape (iPlanet), vous **devriez** avoir besoin de définir un nom de variable d'environnement que PHP utilisera pour voir si tout est correct pour continuer l'exécution.

Note
La définition de cette variable peut avoir des conséquences sur la sécurité. Sachez ce que vous faites avant de faire cela .

fastcgi.impersonate string

FastCGI sous IIS (sur les systèmes d'exploitation basés sur WINNT) supporte la possibilité de déterminer la marque de sécurité du client appelant. Cela permet à IIS de définir le contexte de sécurité sur lequel la requête est exécuté. mod_fastcgi sous Apache ne supporte actuellement pas cette fonctionnalité (03/17/2002). Définie à 1 si vous utilisez IIS. Par défaut, vaut 0.

cgi.rfc2616_headers int

Dit à PHP quel type d'en-tête doit être utilisé lors de l'envoi du code réponse HTTP. Si définie à 0; PHP enverra un en-tête Status: qui est supportée par Apache et les autres serveurs web. Lorsque définie à -1, PHP enverra un en-tête répondant à la spécification de la RFC 2616 . Laissez cette valeur à 0 à moins que vous sachiez ce que vous faites.

11.7.2.6 Chargement de fichiers

Nom	Par défaut	Modifiable	Historique
file_uploads	"1"	PHP_INI_SYSTEM	PHP_INI_ALL en PHP <= 4.2.3. Disponible depuis PHP 4.0.3.
upload_tmp_dir	NULL	PHP_INI_SYSTEM	
upload_max_filesize	"2M"	PHP_INI_PERDIR	PHP_INI_ALL en PHP <= 4.2.3.

Voici un éclaircissement sur l'utilisation des directives de configuration.

file_uploads boolean

Autorise ou non le chargement de fichiers par HTTP. Voir aussi les directives upload_max_filesize , upload_tmp_dir et post_max_size .

Lorsqu'un entier est utilisé, sa valeur est mesuré en octets. Vous pouvez également utiliser la notation sténographique comme décrit dans cette [entrée de la FAQ](#) .

upload_tmp_dir string

Le répertoire temporaire utilisé pour sotcker les fichiers lors du chargement. L'utilisateur sous lequel fonctionne PHP doit avoir les droits en écriture sur ce répertoire. Si non spécifié, PHP utilisera celui par défaut du système.

upload_max_filesize integer

La taille maximale en octets d'un fichier à charger.

Lorsqu'un entier est utilisé, sa valeur est mesuré en octets. Vous pouvez également utiliser la notation sténographique comme décrit dans cette [entrée de la FAQ](#) .

11.7.2.7 SQL général

Nom	Par défaut	Modifiable	Historique
sql.safe_mode	"0"	PHP_INI_SYSTEM	

Voici un éclaircissement sur l'utilisation des directives de configuration.

sql.safe_mode boolean

11.7.2.8 Directives de configuration de débogage

Attention
Seulement PHP 3 implémente un débogueur par défaut ; voir Débogueur PHP pour plus d'informations.

debugger.host string

Nom du DNS ou adresse IP de l'hôte utilisé par le débogueur.

debugger.port string

Numéro du port utilisé par le débogueur.

debugger.enabled boolean

Active ou non le débogueur.

11.8 Liste des Fuseaux Horaires Supportés

11.8.1 Liste des fuseaux horaires dans le groupe Africa (Afrique)

Africa/Abidjan	Africa/Accra	Africa/Addis_Ababa	Africa/Algiers	Africa/Asmera
Africa/Bamako	Africa/Bangui	Africa/Banjul	Africa/Bissau	Africa/Blantyre
Africa/Brazzaville	Africa/Bujumbura	Africa/Cairo	Africa/Casablanca	Africa/Ceuta
Africa/Conakry	Africa/Dakar	Africa/Dar_es_Salaam	Africa/Djibouti	Africa/Douala
Africa/El_Aaiun	Africa/Freetown	Africa/Gaborone	Africa/Harare	Africa/Johannesburg
Africa/Kampala	Africa/Khartoum	Africa/Kigali	Africa/Kinshasa	Africa/Lagos
Africa/Libreville	Africa/Lome	Africa/Luanda	Africa/Lubumbashi	Africa/Lusaka
Africa/Malabo	Africa/Maputo	Africa/Maseru	Africa/Mbabane	Africa/Mogadishu
Africa/Monrovia	Africa/Nairobi	Africa/Ndjamena	Africa/Niamey	Africa/Nouakchott
Africa/Ouagadougou	Africa/Porto-Novo	Africa/Sao_Tome	Africa/Timbuktu	Africa/Tripoli

Africa/Tunis	Africa/Windhoek
--------------	-----------------

11.8.2 Liste des fuseaux horaires dans le groupe America (Amérique)

America/Adak	America/Anchorage	America/Anguilla	Amer
America/Argentina/Buenos_Aires	America/Argentina/Catamarca	America/Argentina/ComodRivadavia	Amer
America/Argentina/La_Rioja	America/Argentina/Mendoza	America/Argentina/Rio_Gallegos	Amer
America/Argentina/Ushuaia	America/Aruba	America/Asuncion	Amer
America/Barbados	America/Belem	America/Belize	Amer
America/Boise	America/Buenos_Aires	America/Cambridge_Bay	Amer
America/Caracas	America/Catamarca	America/Cayenne	Amer
America/Chihuahua	America/Coral_Harbour	America/Cordoba	Amer
America/Curacao	America/Danmarkshavn	America/Dawson	Amer
America/Detroit	America/Dominica	America/Edmonton	Amer
America/Ensenada	America/Fort_Wayne	America/Fortaleza	Amer
America/Goose_Bay	America/Grand_Turk	America/Grenada	Amer
America/Guayaquil	America/Guyana	America/Halifax	Amer
America/Indiana/Indianapolis	America/Indiana/Knox	America/Indiana/Marengo	Amer
America/Inuvik	America/Iqaluit	America/Jamaica	Amer
America/Kentucky/Louisville	America/Kentucky/Monticello	America/Knox_IN	Amer
America/Los_Angeles	America/Louisville	America/Maceio	Amer
America/Martinique	America/Mazatlan	America/Mendoza	Amer
America/Mexico_City	America/Miquelon	America/Monterrey	Amer
America/Montserrat	America/Nassau	America/New_York	Amer
America/Noronha	America/North_Dakota/Center	America/Panama	Amer
America/Phoenix	America/Port-au-Prince	America/Port_of_Spain	Amer
America/Puerto_Rico	America/Rainy_River	America/Rankin_Inlet	Amer
America/Rio_Branco	America/Rosario	America/Santiago	Amer
America/Scoresbysund	America/Shiprock	America/St_Johns	Amer
America/St_Thomas	America/St_Vincent	America/Swift_Current	Amer
America/Thunder_Bay	America/Tijuana	America/Toronto	Amer
America/Virgin	America/Whitehorse	America/Winnipeg	Amer
Brazil/Acre	Brazil/DeNoronha	Brazil/East	Brazil
Canada/Central	Canada/East-Saskatchewan	Canada/Eastern	Canad
Canada/Pacific	Canada/Saskatchewan	Canada/Yukon	Chile
Mexico/BajaNorte	Mexico/BajaSur	Mexico/General	US/A
US/Arizona	US/Central	US/East-Indiana	US/E
US/Indiana-Starke	US/Michigan	US/Mountain	US/P
US/Samoa			

11.8.3 List of timezones in the group Antarctica

Antarctica/Casey	Antarctica/Davis	Antarctica/DumontDUrville	Antarctica/Mawson	Antarctica/McMurdo
Antarctica/Palmer	Antarctica/Rothera	Antarctica/South_Pole	Antarctica/Syowa	Antarctica/Vostok

11.8.4 List of timezones in the group Arctic

Arctic/Longyearbyen

11.8.5 Liste des fuseaux horaires dans le groupe Asia (Asie)

Asia/Aden	Asia/Almaty	Asia/Amman	Asia/Anadyr	Asia/Aqtau
Asia/Aqtobe	Asia/Ashgabat	Asia/Ashkhabad	Asia/Baghdad	Asia/Bahrain
Asia/Baku	Asia/Bangkok	Asia/Beirut	Asia/Bishkek	Asia/Brunei
Asia/Calcutta	Asia/Choibalsan	Asia/Chongqing	Asia/Chungking	Asia/Colombo
Asia/Dacca	Asia/Damascus	Asia/Dhaka	Asia/Dili	Asia/Dubai
Asia/Dushanbe	Asia/Gaza	Asia/Harbin	Asia/Hong_Kong	Asia/Hovd
Asia/Irkutsk	Asia/Istanbul	Asia/Jakarta	Asia/Jayapura	Asia/Jerusalem
Asia/Kabul	Asia/Kamchatka	Asia/Karachi	Asia/Kashgar	Asia/Katmandu
Asia/Krasnoyarsk	Asia/Kuala_Lumpur	Asia/Kuching	Asia/Kuwait	Asia/Macao
Asia/Macau	Asia/Magadan	Asia/Makassar	Asia/Manila	Asia/Muscat
Asia/Nicosia	Asia/Novosibirsk	Asia/Omsk	Asia/Oral	Asia/Phnom_Penh
Asia/Pontianak	Asia/Pyongyang	Asia/Qatar	Asia/Qyzylorda	Asia/Rangoon
Asia/Riyadh	Asia/Saigon	Asia/Sakhalin	Asia/Samarkand	Asia/Seoul
Asia/Shanghai	Asia/Singapore	Asia/Taipei	Asia/Tashkent	Asia/Tbilisi
Asia/Tehran	Asia/Tel_Aviv	Asia/Thimbu	Asia/Thimphu	Asia/Tokyo
Asia/Ujung_Pandang	Asia/Ulaanbaatar	Asia/Ulan_Bator	Asia/Urumqi	Asia/Vientiane
Asia/Vladivostok	Asia/Yakutsk	Asia/Yekaterinburg	Asia/Yerevan	Indian/Antananarivo
Indian/Chagos	Indian/Christmas	Indian/Cocos	Indian/Comoro	Indian/Kerguelen
Indian/Mahe	Indian/Maldives	Indian/Mauritius	Indian/Mayotte	Indian/Reunion

11.8.6 Liste des fuseaux horaires dans le groupe Atlantic (Atlantique)

Atlantic/Azores	Atlantic/Bermuda	Atlantic/Canary	Atlantic/Cape_Verde	Atlantic/Faeroe
Atlantic/Jan_Mayen	Atlantic/Madeira	Atlantic/Reykjavik	Atlantic/South_Georgia	Atlantic/St_Helena
Atlantic/Stanley				

11.8.7 List of timezones in the group Australia

Australia/ACT	Australia/Adelaide	Australia/Brisbane	Australia/Broken_Hill	Australia/Canberra
Australia/Currie	Australia/Darwin	Australia/Hobart	Australia/LHI	Australia/Lindeman
Australia/Lord_Howe	Australia/Melbourne	Australia/North	Australia/NSW	Australia/Perth
Australia/Queensland	Australia/South	Australia/Sydney	Australia/Tasmania	Australia/Victoria
Australia/West	Australia/Yancowinna			

11.8.8 Liste des fuseaux horaires dans le groupe Europe

Europe/Amsterdam	Europe/Andorra	Europe/Athens	Europe/Belfast	Europe/Belgrade
Europe/Berlin	Europe/Bratislava	Europe/Brussels	Europe/Bucharest	Europe/Budapest
Europe/Chisinau	Europe/Copenhagen	Europe/Dublin	Europe/Gibraltar	Europe/Helsinki
Europe/Istanbul	Europe/Kaliningrad	Europe/Kiev	Europe/Lisbon	Europe/Ljubljana

Europe/London	Europe/Luxembourg	Europe/Madrid	Europe/Malta	Europe/Mariehamn
Europe/Minsk	Europe/Monaco	Europe/Moscow	Europe/Nicosia	Europe/Oslo
Europe/Paris	Europe/Prague	Europe/Riga	Europe/Rome	Europe/Samara
Europe/San_Marino	Europe/Sarajevo	Europe/Simferopol	Europe/Skopje	Europe/Sofia
Europe/Stockholm	Europe/Tallinn	Europe/Tirane	Europe/Tiraspol	Europe/Uzhgorod
Europe/Vaduz	Europe/Vatican	Europe/Vienna	Europe/Vilnius	Europe/Warsaw
Europe/Zagreb	Europe/Zaporozhye	Europe/Zurich		

11.8.9 Liste des fuseaux horaires dans le groupe Pacific (Pacifique)

Pacific/Apia	Pacific/Auckland	Pacific/Chatham	Pacific/Easter	Pacific/Efate
Pacific/Enderbury	Pacific/Fakaofu	Pacific/Fiji	Pacific/Funafuti	Pacific/Galapagos
Pacific/Gambier	Pacific/Guadalcanal	Pacific/Guam	Pacific/Honolulu	Pacific/Johnston
Pacific/Kiritimati	Pacific/Kosrae	Pacific/Kwajalein	Pacific/Majuro	Pacific/Marquesas
Pacific/Midway	Pacific/Nauru	Pacific/Niue	Pacific/Norfolk	Pacific/Noumea
Pacific/Pago_Pago	Pacific/Palau	Pacific/Pitcairn	Pacific/Ponape	Pacific/Port_Moresby
Pacific/Rarotonga	Pacific/Saipan	Pacific/Samoa	Pacific/Tahiti	Pacific/Tarawa
Pacific/Tongatapu	Pacific/Truk	Pacific/Wake	Pacific/Wallis	Pacific/Yap

11.8.10 Liste des fuseaux horaires dans le groupe Autres

CET	CST6CDT	Cuba	EET	Egypt
Eire	EST	EST5EDT	Etc/GMT	Etc/GMT+0
Etc/GMT+1	Etc/GMT+10	Etc/GMT+11	Etc/GMT+12	Etc/GMT+2
Etc/GMT+3	Etc/GMT+4	Etc/GMT+5	Etc/GMT+6	Etc/GMT+7
Etc/GMT+8	Etc/GMT+9	Etc/GMT-0	Etc/GMT-1	Etc/GMT-10
Etc/GMT-11	Etc/GMT-12	Etc/GMT-13	Etc/GMT-14	Etc/GMT-2
Etc/GMT-3	Etc/GMT-4	Etc/GMT-5	Etc/GMT-6	Etc/GMT-7
Etc/GMT-8	Etc/GMT-9	Etc/GMT0	Etc/Greenwich	Etc/UCT
Etc/Universal	Etc/UTC	Etc/Zulu	Factory	GB
GB-Eire	GMT	GMT+0	GMT-0	GMT0
Greenwich	Hongkong	HST	Iceland	Iran
Israel	Jamaica	Japan	Kwajalein	Libya
MET	MST	MST7MDT	Navajo	NZ
NZ-CHAT	Poland	Portugal	PRC	PST8PDT
ROC	ROK	Singapore	Turkey	UCT
Universal	UTC	W-SU	WET	Zulu

11.9 extensions

extensions.membership

extensions.membership.bundled**extensions.membership.external****extensions.membership.pecl****extensions.state****extensions.state.experimental**

11.10 Liste des alias

Il y a beaucoup de fonctions en PHP qui peuvent être appelées sous de différents noms. Dans la plus part des cas, il n'y pas de nom préféré par rapport à un autre, `is_int` et `is_integer` sont exactement identiques, par exemple. Ces modifications de noms ont été réalisées en général à cause d'un changement dans l'API d'origine ou pour d'autres raisons et les anciens noms sont conservés uniquement pour des raisons de compatibilité ascendante. C'est une très mauvaise habitude d'utiliser ces alias, car ils risquent à tous moment de disparaître, rendus obsolète sans préavis, ou bien par un simple changement de nom, ce qui rend votre script inutilisable avec des versions plus récentes de PHP. Préférez toujours les versions officielles. En fait, cette liste est surtout destinée à ceux qui doivent mettre à jour leurs scripts avec les syntaxes plus récentes.

Cette liste est en phase avec PHP 4.0.6. Pour une liste d'alias qui soit mise à jour quotidiennement, voyez http://zend.com/phpfunc/all_aliases.php .

Alias	Fonction principale	Extension mère
<code>_</code>	<code>gettext</code>	<code>Gettext</code>
<code>add</code>	<code>swfmovie_add</code>	<code>Ming (flash)</code>
<code>add</code>	<code>swfsprite_add</code>	<code>Ming (flash)</code>
<code>add_root</code>	<code>domxml_add_root</code>	<code>DOM XML</code>
<code>addaction</code>	<code>swfbutton_addAction</code>	<code>Ming (flash)</code>
<code>addcolor</code>	<code>swfdisplayitem_addColor</code>	<code>Ming (flash)</code>
<code>addentry</code>	<code>swfgradient_addEntry</code>	<code>Ming (flash)</code>
<code>addfill</code>	<code>swfshape_addfill</code>	<code>Ming (flash)</code>
<code>addshape</code>	<code>swfbutton_addShape</code>	<code>Ming (flash)</code>
<code>addstring</code>	<code>swftext_addString</code>	<code>Ming (flash)</code>
<code>addstring</code>	<code>swftextfield_addString</code>	<code>Ming (flash)</code>
<code>align</code>	<code>swftextfield_align</code>	<code>Ming (flash)</code>
<code>attributes</code>	<code>domxml_attributes</code>	<code>DOM XML</code>
<code>children</code>	<code>domxml_children</code>	<code>DOM XML</code>
<code>chop</code>	<code>rtrim</code>	<code>Base syntax</code>
<code>close</code>	<code>closedir</code>	<code>Base syntax</code>
<code>com_get</code>	<code>com_propget</code>	<code>COM</code>
<code>com_propset</code>	<code>com_propput</code>	<code>COM</code>

com_set	com_propput	COM
cv_add	ccvs_add	CCVS
cv_auth	ccvs_auth	CCVS
cv_command	ccvs_command	CCVS
cv_count	ccvs_count	CCVS
cv_delete	ccvs_delete	CCVS
cv_done	ccvs_done	CCVS
cv_init	ccvs_init	CCVS
cv_lookup	ccvs_lookup	CCVS
cv_new	ccvs_new	CCVS
cv_report	ccvs_report	CCVS
cv_return	ccvs_return	CCVS
cv_reverse	ccvs_reverse	CCVS
cv_sale	ccvs_sale	CCVS
cv_status	ccvs_status	CCVS
cv_textvalue	ccvs_textvalue	CCVS
cv_void	ccvs_void	CCVS
die	exit	Miscellaneous functions
dir	getdir	Base syntax
diskfreespace	disk_free_space	Filesystem
domxml_getattr	domxml_get_attribute	DOM XML
domxml_setattr	domxml_set_attribute	DOM XML
doubleval	floatval	Base syntax
drawarc	swfshape_drawarc	Ming (flash)
drawcircle	swfshape_drawcircle	Ming (flash)
drawcubic	swfshape_drawcubic	Ming (flash)
drawcubicto	swfshape_drawcubicto	Ming (flash)
drawcurve	swfshape_drawcurve	Ming (flash)
drawcurveto	swfshape_drawcurveto	Ming (flash)
drawglyph	swfshape_drawglyph	Ming (flash)
drawline	swfshape_drawline	Ming (flash)
drawlineto	swfshape_drawlineto	Ming (flash)
dtd	domxml_intdtd	DOM XML
dumpmem	domxml_dumpmem	DOM XML
fbsql	fbsql_db_query	FrontBase
fputs	fwrite	Base syntax
get_attribute	domxml_get_attribute	DOM XML
getascent	swffont_getAscent	Ming (flash)
getascent	swftext_getAscent	Ming (flash)
getattr	domxml_get_attribute	DOM XML
getdescent	swffont_getDescent	Ming (flash)
getdescent	swftext_getDescent	Ming (flash)
getheight	swfbitmap_getHeight	Ming (flash)
getleading	swffont_getLeading	Ming (flash)
getleading	swftext_getLeading	Ming (flash)

getshape1	swfmorph_getShape1	Ming (flash)
getshape2	swfmorph_getShape2	Ming (flash)
getwidth	swfbitmap_getWidth	Ming (flash)
getwidth	swffont_getWidth	Ming (flash)
getwidth	swftext_getWidth	Ming (flash)
gzputs	gzwrite	Zlib
i18n_convert	mb_convert_encoding	Multi-bytes Strings
i18n_discover_encoding	mb_detect_encoding	Multi-bytes Strings
i18n_http_input	mb_http_input	Multi-bytes Strings
i18n_http_output	mb_http_output	Multi-bytes Strings
i18n_internal_encoding	mb_internal_encoding	Multi-bytes Strings
i18n_ja_jp_hantozen	mb_convert_kana	Multi-bytes Strings
i18n_mime_header_decode	mb_decode_mimeheader	Multi-bytes Strings
i18n_mime_header_encode	mb_encode_mimeheader	Multi-bytes Strings
imap_create	imap_createmailbox	IMAP
imap_fetchtext	imap_body	IMAP
imap_getmailboxes	imap_list_full	IMAP
imap_getsubscribed	imap_lsub_full	IMAP
imap_header	imap_headerinfo	IMAP
imap_listmailbox	imap_list	IMAP
imap_listsubscribed	imap_lsub	IMAP
imap_rename	imap_renamemailbox	IMAP
imap_scan	imap_listscan	IMAP
imap_scanmailbox	imap_listscan	IMAP
ini_alter	ini_set	Base syntax
is_double	is_float	Base syntax
is_integer	is_int	Base syntax
is_long	is_int	Base syntax
is_real	is_float	Base syntax
is_writeable	is_writable	Base syntax
join	implode	Base syntax
labelframe	swfmovie_labelFrame	Ming (flash)
labelframe	swfsprite_labelFrame	Ming (flash)
last_child	domxml_last_child	DOM XML
lastchild	domxml_last_child	DOM XML
ldap_close	ldap_unbind	LDAP
magic_quotes_runtime	set_magic_quotes_runtime	Base syntax
mbstrcut	mb_strcut	Multi-bytes Strings
mbstrlen	mb_strlen	Multi-bytes Strings
mbstrpos	mb_strpos	Multi-bytes Strings
mbstrrpos	mb_strrpos	Multi-bytes Strings
mbsubstr	mb_substr	Multi-bytes Strings
ming_setcubicthreshold	ming setCubicThreshold	Ming (flash)
ming_setscale	ming setScale	Ming (flash)
move	swfdisplayitem_move	Ming (flash)

movepen	swfshape_movepen	Ming (flash)
movepento	swfshape_movepento	Ming (flash)
moveto	swfdisplayitem_moveTo	Ming (flash)
moveto	swffill_moveTo	Ming (flash)
moveto	swftext_moveTo	Ming (flash)
msql	msql_db_query	mSQL
msql_createdb	msql_create_db	mSQL
msql_dbname	msql_result	mSQL
msql_dropdb	msql_drop_db	mSQL
msql_fieldflags	msql_field_flags	mSQL
msql_fieldlen	msql_field_len	mSQL
msql_fieldname	msql_field_name	mSQL
msql_fieldtable	msql_field_table	mSQL
msql_fieldtype	msql_field_type	mSQL
msql_freeresult	msql_free_result	mSQL
msql_listdbs	msql_list_dbs	mSQL
msql_listfields	msql_list_fields	mSQL
msql_listtables	msql_list_tables	mSQL
msql_numfields	msql_num_fields	mSQL
msql_numrows	msql_num_rows	mSQL
msql_regcase	sql_regcase	mSQL
msql_selectdb	msql_select_db	mSQL
msql_tablename	msql_result	mSQL
mssql_affected_rows	sybase_affected_rows	Sybase
mssql_affected_rows	sybase_affected_rows	Sybase
mssql_close	sybase_close	Sybase
mssql_close	sybase_close	Sybase
mssql_connect	sybase_connect	Sybase
mssql_connect	sybase_connect	Sybase
mssql_data_seek	sybase_data_seek	Sybase
mssql_data_seek	sybase_data_seek	Sybase
mssql_fetch_array	sybase_fetch_array	Sybase
mssql_fetch_array	sybase_fetch_array	Sybase
mssql_fetch_field	sybase_fetch_field	Sybase
mssql_fetch_field	sybase_fetch_field	Sybase
mssql_fetch_object	sybase_fetch_object	Sybase
mssql_fetch_object	sybase_fetch_object	Sybase
mssql_fetch_row	sybase_fetch_row	Sybase
mssql_fetch_row	sybase_fetch_row	Sybase
mssql_field_seek	sybase_field_seek	Sybase
mssql_field_seek	sybase_field_seek	Sybase
mssql_free_result	sybase_free_result	Sybase
mssql_free_result	sybase_free_result	Sybase
mssql_get_last_message	sybase_get_last_message	Sybase
mssql_get_last_message	sybase_get_last_message	Sybase

mssql_min_client_severity	sybase_min_client_severity	Sybase
mssql_min_error_severity	sybase_min_error_severity	Sybase
mssql_min_message_severity	sybase_min_message_severity	Sybase
mssql_min_server_severity	sybase_min_server_severity	Sybase
mssql_num_fields	sybase_num_fields	Sybase
mssql_num_fields	sybase_num_fields	Sybase
mssql_num_rows	sybase_num_rows	Sybase
mssql_num_rows	sybase_num_rows	Sybase
mssql_pconnect	sybase_pconnect	Sybase
mssql_pconnect	sybase_pconnect	Sybase
mssql_query	sybase_query	Sybase
mssql_query	sybase_query	Sybase
mssql_result	sybase_result	Sybase
mssql_result	sybase_result	Sybase
mssql_select_db	sybase_select_db	Sybase
mssql_select_db	sybase_select_db	Sybase
multicolor	swfdisplayitem_multColor	Ming (flash)
mysql	mysql_db_query	MySQL
mysql_createdb	mysql_create_db	MySQL
mysql_db_name	mysql_result	MySQL
mysql_dbname	mysql_result	MySQL
mysql_dropdb	mysql_drop_db	MySQL
mysql_fieldflags	mysql_field_flags	MySQL
mysql_fieldlen	mysql_field_len	MySQL
mysql_fieldname	mysql_field_name	MySQL
mysql_fieldtable	mysql_field_table	MySQL
mysql_fieldtype	mysql_field_type	MySQL
mysql_freeresult	mysql_free_result	MySQL
mysql_listdbs	mysql_list_dbs	MySQL
mysql_listfields	mysql_list_fields	MySQL
mysql_listtables	mysql_list_tables	MySQL
mysql_numfields	mysql_num_fields	MySQL
mysql_numrows	mysql_num_rows	MySQL
mysql_selectdb	mysql_select_db	MySQL
mysql_tablename	mysql_result	MySQL
name	domxml_attrname	DOM XML
new_child	domxml_new_child	DOM XML
new_xmldoc	domxml_new_xmldoc	DOM XML
nextframe	swfmovie_nextFrame	Ming (flash)
nextframe	swfsprite_nextFrame	Ming (flash)
node	domxml_node	DOM XML
oci8append	ocicollappend	OCI8
oci8assign	ocicollassign	OCI8
oci8assignelem	ocicollassignelem	OCI8
oci8close	ocicloselob	OCI8

oci8free	ocifreecoll	OCI8
oci8free	ocifreedesc	OCI8
oci8getelem	ocicollgetelem	OCI8
oci8load	ociloadlob	OCI8
oci8max	ocicollmax	OCI8
oci8ocifreecursor	ocifreestatement	OCI8
oci8save	ocisavelob	OCI8
oci8savefile	ocisavelobfile	OCI8
oci8size	ocicollsize	OCI8
oci8trim	ocicolltrim	OCI8
oci8writetemporary	ociwritetemporarylob	OCI8
oci8writetofile	ociwritelobtofile	OCI8
odbc_do	odbc_exec	ODBC
odbc_field_precision	odbc_field_len	ODBC
output	swfmovie_output	Ming (flash)
parent	domxml_parent	DOM XML
pdf_add_outline	pdf_add_bookmark	PDF
pg_clientencoding	pg_client_encoding	PostgreSQL
pg_setclientencoding	pg_set_client_encoding	PostgreSQL
pos	current	Base syntax
recode	recode_string	Recode
remove	swfmovie_remove	Ming (flash)
remove	swfsprite_remove	Ming (flash)
rewind	rewinddir	Base syntax
root	domxml_root	DOM XML
rotate	swfdisplayitem_rotate	Ming (flash)
rotateto	swfdisplayitem_rotateTo	Ming (flash)
rotateto	swffill_rotateTo	Ming (flash)
save	swfmovie_save	Ming (flash)
savetofile	swfmovie_saveToFile	Ming (flash)
scale	swfdisplayitem_scale	Ming (flash)
scaleto	swfdisplayitem_scaleTo	Ming (flash)
scaleto	swffill_scaleTo	Ming (flash)
set_attribute	domxml_set_attribute	DOM XML
set_content	domxml_set_content	DOM XML
setaction	swfbutton_setAction	Ming (flash)
setattr	domxml_set_attribute	DOM XML
setbackground	swfmovie_setBackground	Ming (flash)
setbounds	swftextfield_setBounds	Ming (flash)
setcolor	swftext_setColor	Ming (flash)
setcolor	swftextfield_setColor	Ming (flash)
setdepth	swfdisplayitem_setDepth	Ming (flash)
setdimension	swfmovie_setDimension	Ming (flash)
setdown	swfbutton_setDown	Ming (flash)
setfont	swftext_setFont	Ming (flash)

setfont	swftextfield_setFont	Ming (flash)
setframes	swfmovie_setFrames	Ming (flash)
setframes	swfsprite_setFrames	Ming (flash)
setheight	swftext_setHeight	Ming (flash)
setheight	swftextfield_setHeight	Ming (flash)
sethit	swfbutton_setHit	Ming (flash)
setindentation	swftextfield_setIndentation	Ming (flash)
setleftfill	swfshape_setleftfill	Ming (flash)
setleftmargin	swftextfield_setLeftMargin	Ming (flash)
setline	swfshape_setline	Ming (flash)
setlinespacing	swftextfield_setLineSpacing	Ming (flash)
setmargins	swftextfield_setMargins	Ming (flash)
setmatrix	swfdisplayitem_setMatrix	Ming (flash)
setname	swfdisplayitem_setName	Ming (flash)
setname	swftextfield_setName	Ming (flash)
setover	swfbutton_setOver	Ming (flash)
setrate	swfmovie_setRate	Ming (flash)
setratio	swfdisplayitem_setRatio	Ming (flash)
setrightfill	swfshape_setrightfill	Ming (flash)
setrightmargin	swftextfield_setRightMargin	Ming (flash)
setspacing	swftext_setSpacing	Ming (flash)
setup	swfbutton_setUp	Ming (flash)
show_source	highlight_file	Base syntax
sizeof	count	Base syntax
skewx	swfdisplayitem_skewX	Ming (flash)
skewxto	swfdisplayitem_skewXTo	Ming (flash)
skewxto	swffill_skewXTo	Ming (flash)
skewy	swfdisplayitem_skewy	Ming (flash)
skewyto	swfdisplayitem_skewYTo	Ming (flash)
skewyto	swffill_skewYTo	Ming (flash)
snmpwalkoid	snmprealwalk	SNMP
strchr	strstr	Base syntax
streammp3	swfmovie_streamMp3	Ming (flash)
swfaction	swfaction_init	Ming (flash)
swfbitmap	swfbitmap_init	Ming (flash)
swfbutton	swfbutton_init	Ming (flash)
swffill	swffill_init	Ming (flash)
swffont	swffont_init	Ming (flash)
swfgradient	swfgradient_init	Ming (flash)
swfmorph	swfmorph_init	Ming (flash)
swfmovie	swfmovie_init	Ming (flash)
swfshape	swfshape_init	Ming (flash)
swfsprite	swfsprite_init	Ming (flash)
swftext	swftext_init	Ming (flash)
swftextfield	swftextfield_init	Ming (flash)

<code>unlink</code>	<code>domxml_unlink_node</code>	DOM XML
<code>xptr_new_context</code>	<code>xpath_new_context</code>	DOM XML

11.11 Mots réservés en PHP

Cette annexe est une liste d'identifiants pré-définis en PHP. Aucun des identifiants utilisés ici ne doit être repris comme nom de variable ou de fonction dans vos scripts. Ces identifiants incluent des mots clés, des constantes, des classes, et des variables pré-définies. Ces listes ne sont pas complètes ou exhaustives.

11.11.1 Liste de mots-clé

Ces mots ont un sens spécial en PHP. Certains représentent des objets qui ressemblent à des fonctions, d'autres à des constantes, et ainsi de suite, mais ils n'en sont pas vraiment : ce sont des structures de langage. Vous ne pourrez pas les utiliser comme constantes, nom de classe, nom de fonctions ou nom de méthodes. Vous pouvez les utiliser comme noms de variables, mais cela risque de vous mener des confusions.

<code>and</code>	<code>or</code>	<code>xor</code>	<code>__FILE__</code>	<code>exception</code> (PHP 5)
<code>__LINE__</code>	<code>array</code>	<code>as</code>	<code>break</code>	<code>case</code>
<code>class</code>	<code>const</code>	<code>continue</code>	<code>declare</code>	<code>default</code>
<code>die</code>	<code>do</code>	<code>echo</code>	<code>else</code>	<code>elseif</code>
<code>empty</code>	<code>enddeclare</code>	<code>endfor</code>	<code>endforeach</code>	<code>endif</code>
<code>endswitch</code>	<code>endwhile</code>	<code>eval</code>	<code>exit</code>	<code>extends</code>
<code>for</code>	<code>foreach</code>	<code>function</code>	<code>global</code>	<code>if</code>
<code>include</code>	<code>include_once</code>	<code>isset</code>	<code>list</code>	<code>new</code>
<code>print</code>	<code>require</code>	<code>require_once</code>	<code>return</code>	<code>static</code>
<code>switch</code>	<code>unset</code>	<code>use</code>	<code>var</code>	<code>while</code>
<code>__FUNCTION__</code>	<code>__CLASS__</code>	<code>__METHOD__</code>	<code>final</code> (PHP 5)	<code>php_user_filter</code> (PHP 5)
<code>interface</code> (PHP 5)	<code>implements</code> (PHP 5)	<code>extends</code>	<code>public</code> (PHP 5)	<code>private</code> (PHP 5)
<code>protected</code> (PHP 5)	<code>abstract</code> (PHP 5)	<code>clone</code> (PHP 5)	<code>try</code> (PHP 5)	<code>catch</code> (PHP 5)
<code>throw</code> (PHP 5)	<code>cfunction</code> (PHP 4 uniquement)	<code>old_function</code> (PHP 4 uniquement)		

11.11.2 Variables pré-définies

Depuis PHP 4.1.0, la méthode recommandée pour accéder à des variables externes est d'utiliser les super-globales mentionnées ci-dessous. Avant cela, il fallait utiliser soit l'option de configuration `register_globals`, soit les tableaux pré-définis `$HTTP_*_VARS`. Depuis PHP 5.0.0, les tableaux prédéfinis PHP peuvent être désactivés avec l'option de configuration `register_long_arrays`.

11.11.2.1 Variables de serveur : `$_SERVER`

Note
Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait <code>\$HTTP_SERVER_VARS</code> .

`$_SERVER` est un tableau contenant des informations comme les en-têtes, dossiers et chemins du script. Les entrées de ce tableau sont créées par le serveur web. Il n'y a aucune garantie que tous les serveurs les rempliront tous; certains en oublieront quelques unes et en rajouteront de nouvelles non mentionnées ici. Cependant, un grand nombre de ces variables fait partie des spécifications CGI 1.1 , et vous pouvez donc vous attendre à les retrouver.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global `$_SERVER`; pour y accéder, comme vous le faisiez avec `$HTTP_SERVER_VARS` .

`$HTTP_SERVER_VARS` contient les mêmes informations, mais n'est pas autoglobale. (Notez que `$HTTP_SERVER_VARS` et `$_SERVER` sont des variables différentes et que PHP les traite en tant que tel)

Si la directive `register_globals` est active, alors ces variables seront aussi rendues directement accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux `$_SERVER` et `$HTTP_SERVER_VARS` . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre Utiliser la directive Register Globals . Ces variables globales individuelles ne sont pas des globales.

Vous pouvez éventuellement trouver les éléments suivants dans la variable `$_SERVER` . Notez que certains, n'auront pas de sens si vous utilisez PHP en ligne de commande.

' `PHP_SELF` '

Le nom du fichier du script en cours d'exécution, par rapport à la racine web. Par exemple, `$_SERVER['PHP_SELF']` dans le script situé à l'adresse `http://www.monsite.com/test.php/foo.bar` sera `/test.php/foo.bar` . La constante `__FILE__` contient le chemin complet ainsi que le nom du fichier (i.e. inclut) courant.

Si PHP fonctionne en ligne de commande, cette variable contient le nom du script depuis PHP 4.3.0. Dans les versions antérieures, cette variable n'était pas disponible.

' `argv` '

Tableau des arguments passés au script. Lorsque le script est appelé en ligne de commande, cela donne accès aux arguments, comme en langage C. Lorsque le script est appelé avec la méthode GET, ce tableau contiendra la chaîne de requête.

' `argc` '

Contient le nombre de paramètres de la ligne de commande passés au script (si le script fonctionne en ligne de commande).

' `GATEWAY_INTERFACE` '

Numéro de révision de l'interface CGI du serveur : i.e. ' `CGI/1.1` '.

' `SERVER_NAME` '

Le nom du serveur hôte qui exécute le script suivant. Si le script est exécuté sur un hôte virtuel, ce sera la valeur définie pour cet hôte virtuel.

' `SERVER_SOFTWARE` '

Chaîne d'identification du serveur, qui est donnée dans les en-têtes lors de la réponse aux requêtes.

' `SERVER_PROTOCOL` '

Nom et révision du protocole de communication : i.e. ' `HTTP/1.0` ';

' `REQUEST_METHOD` '

Méthode de requête utilisée pour accéder à la page; i.e. ' `GET` ', ' `HEAD` ', ' `POST` ', ' `PUT` '.

Note

Le script PHP se termine après avoir envoyé les en-têtes (après avoir produit n'importe quelle sortie sans avoir affiché le buffer) si la méthode de la requête était HEAD .

- ' REQUEST_TIME '
Le temps Unix depuis le début de la requête. Disponible depuis PHP 5.1.0.
- ' QUERY_STRING '
La chaîne de requête, si elle existe, qui est utilisée pour accéder à la page.
- ' DOCUMENT_ROOT '
La racine sous laquelle le script courant est exécuté, comme défini dans la configuration du serveur.
- ' HTTP_ACCEPT '
Contenu de l'en-tête Accept: de la requête courante, s'il y en a une.
- ' HTTP_ACCEPT_CHARSET '
Contenu de l'en-tête Accept-Charset: de la requête courante, s'elle existe. Par exemple : ' iso-8859-1,* ,utf-8 '.
- ' HTTP_ACCEPT_ENCODING '
Contenu de l'en-tête Accept-Encoding: de la requête courante, si elle existe. Par exemple : ' gzip '.
- ' HTTP_ACCEPT_LANGUAGE '
Contenu de l'en-tête Accept-Language: de la requête courante, si elle existe. Par exemple : ' fr '.
- ' HTTP_CONNECTION '
Contenu de l'en-tête Connection: de la requête courante, si elle existe. Par exemple : ' Keep-Alive '.
- ' HTTP_HOST '
Contenu de l'en-tête Host: de la requête courante, si elle existe.
- ' HTTP_REFERER '
L'adresse de la page (si elle existe) qui a conduit le client à la page courante. Cette valeur est affectée par le client, et tous les clients ne le font pas. Certains navigateur permettent même de modifier la valeur de HTTP_REFERER , sous forme de fonctionnalité. En bref, ce n'est pas une valeur de confiance.
- ' HTTP_USER_AGENT '
Contenu de l'en-tête User-Agent: de la requête courante, si elle existe. C'est une chaîne qui décrit le client HTML utilisé pour voir la page courante. Par exemple : Mozilla/4.5 [en] (X11; U; Linux 2.2.9 i586) . Entre autres choses, vous pouvez utiliser cette valeur avec [get_browser](#) pour optimiser votre page en fonction des capacités du client.
- ' HTTPS '
Définissez à une valeur non-vide si le script nécessite d'utiliser le protocole HTTPS.
- ' REMOTE_ADDR '
L'adresse IP du client qui demande la page courante.
- ' REMOTE_HOST '
Le nom de l'hôte qui lit le script courant. La résolution DNS inverse est basée sur la valeur de REMOTE_ADDR .

Note
Votre serveur web doit être configuré pour créer cette variable. Par exemple, pour Apache, vous devez ajouter la directive HostnameLookups On dans le fichier httpd.conf , pour que cette variable existe. Voyez aussi [gethostbyaddr](#) .

- ' REMOTE_PORT '
Le port utilisé par la machine cliente pour communiquer avec le serveur web.
- ' SCRIPT_FILENAME '
Le chemin absolu vers le fichier contenant le script en cours d'exécution.

Note

Si un script est exécuté avec le CLI, avec un chemin relatif, comme file.php ou ../file.php , \$_SERVER['SCRIPT_FILENAME'] contiendra le chemin relatif spécifié par l'utilisateur.

' SERVER_ADMIN '

La valeur donnée à la directive SERVER_ADMIN (pour Apache), dans le fichier de configuration. Si le script est exécuté par un hôte virtuel, ce sera la valeur définie par l'hôte virtuel.

' SERVER_PORT '

Le port de la machine serveur utilisé pour les communications. Par défaut, c'est '80'. En utilisant SSL, par exemple, il sera remplacé par le numéro de port HTTP sécurisé.

' SERVER_SIGNATURE '

Chaîne contenant le numéro de version du serveur et le nom d'hôte virtuel, qui sont ajoutés aux pages générées par le serveur, si cette option est activée.

' PATH_TRANSLATED '

Chemin dans le système de fichier (pas le document-root) jusqu'au script courant, une fois que le serveur a fait une traduction chemin virtuel -> réel.

Note

Depuis PHP 4.3.2, la variable **PATH_TRANSLATED** n'est plus seulement définie implicitement sous Apache 2 SAPI contrairement à la situation sous Apache 1 où elle est définie avec la même valeur que la variable serveur **SCRIPT_FILENAME** lorsqu'elle n'est pas fournie par Apache. Ce changement a été effectué pour être conforme aux spécifications CGI qui fait que la variable **PATH_TRANSLATED** ne doit exister seulement si la variable **PATH_INFO** est définie.

Les utilisateurs d'Apache 2 devrait utiliser AcceptPathInfo = On au lieu de httpd.conf pour définir **PATH_INFO** .

' SCRIPT_NAME '

Contient le nom du script courant. Cela sert lorsque les pages doivent s'appeler elles-mêmes. La constante __FILE__ contient le chemin complet ainsi que le nom du fichier (i.e. inclut) courant.

' REQUEST_URI '

L'URI qui a été fourni pour accéder à cette page. Par exemple : '/index.html'.

' PHP_AUTH_DIGEST '

Lorsque vous utilisez PHP avec Apache en tant que module faisant une identification HTTP Digest , cette variable est définie dans l'en-tête 'Authorization' envoyé par le client (que vous devez donc utiliser pour réaliser la validation appropriée).

' PHP_AUTH_USER '

Lorsque vous utilisez PHP avec Apache ou IIS (ISAPI en PHP 5) en tant que module faisant une identification HTTP, cette variable est définie à l'utilisateur fourni par l'utilisateur.

' PHP_AUTH_PW '

Lorsque vous utilisez PHP avec Apache ou IIS (ISAPI en PHP 5) en tant que module faisant une identification HTTP, cette variable est définie au mot de passe fourni par l'utilisateur.

' AUTH_TYPE '

Lorsque vous utilisez PHP avec Apache en tant que module faisant une identification HTTP, cette variable est définie au type d'identification.

11.11.2.2 Variables d'environnement : \$_ENV

Note

Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait `$HTTP_ENV_VARS` .

Cette variable est importée dans l'espace de nom global de PHP, depuis l'environnement dans lequel l'exécutable PHP fonctionne. De nombreuses valeurs sont fournies par le shell qui exécute PHP, et différents systèmes pouvant disposer de différents shells, même un début de liste serait ici impossible. Reportez-vous à la documentation de votre shell pour connaître une liste de variables pré-définies.

Les autres variables d'environnement incluent les variables CGI, placées ici, indépendamment du fait que PHP fonctionne en tant que CGI ou bien que module du serveur.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global `$_ENV`; pour y accéder, comme vous le faisiez avec `$HTTP_ENV_VARS` .

`$HTTP_ENV_VARS` contient les mêmes informations, mais n'est pas autoglobale. (Notez que `$HTTP_ENV_VARS` et `$_ENV` sont des variables différentes et que PHP les traite comme tel)

Si la directive `register_globals` est active, alors ces variables seront aussi rendues directement accessibles dans le contexte d'exécution global. C'est à dire séparément des tableaux `$_ENV` et `$HTTP_ENV_VARS` . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre [Utiliser la directive Register Globals](#) . Ces variables globales individuelles ne sont pas des globales.

11.11.2.3 HTTP Cookies : `$_COOKIE`

Note

Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait `$HTTP_COOKIE_VARS` .

Un tableau associatif des valeurs passées au script courant via les cookies HTTP. Elle est automatiquement globale dans tous les contextes d'exécution.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global `$_COOKIE`; pour y accéder, comme vous le faisiez avec `$HTTP_COOKIE_VARS` .

`$HTTP_COOKIE_VARS` contient les mêmes informations, mais n'est pas autoglobale. (Notez que `$HTTP_COOKIE_VARS` et `$_COOKIE` sont des variables différentes et que PHP les traite comme tel)

Si la directive `register_globals` est active, alors ces variables seront aussi rendues directement accessibles dans le contexte d'exécution global. C'est à dire séparément des tableaux `$_COOKIE` et `$HTTP_COOKIE_VARS` . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre [Utiliser la directive Register Globals](#) . Ces variables globales individuelles ne sont pas des globales.

11.11.2.4 HTTP GET variables : `$_GET`

Note

Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait `$HTTP_GET_VARS` .

Un tableau associatif des valeurs passées au script courant via le protocole HTTP et la méthode GET. Elle est automatiquement globale dans tous les contextes d'exécution.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global `$_GET`; pour y accéder, comme vous le faisiez avec `$HTTP_GET_VARS`.

`$HTTP_GET_VARS` contient les mêmes informations, mais n'est pas autoglobale. (Notez que `$HTTP_GET_VARS` et `$_GET` sont des variables différentes et que PHP les traite comme tel)

Si la directive `register_globals` est active, alors ces variables seront aussi rendues directement accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux `$_GET` et `$HTTP_GET_VARS`. Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre [Utiliser la directive Register Globals](#). Ces variables globales individuelles ne sont pas des globales.

11.11.2.5 HTTP POST variables : `$_POST`

Note

Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait <code>\$HTTP_POST_VARS</code> .
--

Un tableau associatif des valeurs passées au script courant via le protocole HTTP et la méthode POST. Elle est automatiquement globale dans tous les contextes d'exécution.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global `$_POST`; pour y accéder, comme vous le faisiez avec `$HTTP_POST_VARS`.

`$HTTP_POST_VARS` contient les mêmes informations, mais n'est pas autoglobale. (Notez que `$HTTP_POST_VARS` et `$_POST` sont des variables différentes et que PHP les traite comme tel)

Si la directive `register_globals` est active, alors ces variables seront aussi rendues directement accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux `$_POST` et `$HTTP_POST_VARS`. Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre [Utiliser la directive Register Globals](#). Ces variables globales individuelles ne sont pas des globales.

11.11.2.6 Variable de téléchargement de fichier via HTTP : `$_FILES`

Note

Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait <code>\$HTTP_POST_FILES</code> .

Un tableau associatif des valeurs téléchargées au script courant via le protocole HTTP et la méthode POST. Elle est automatiquement globale dans tous les contextes d'exécution.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global `$_FILES`; pour y accéder, comme vous le faisiez avec `$HTTP_POST_FILES`.

`$HTTP_POST_FILES` contient les mêmes informations, mais n'est pas autoglobale. (Notez que `$HTTP_POST_FILES` et `$_FILES` sont des variables différentes et que PHP les traite comme tel)

Si la directive `register_globals` est active, alors ces variables seront aussi rendues directement accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux `$_FILES` et `$HTTP_POST_FILES`. Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre [Utiliser la directive Register Globals](#). Ces variables globales individuelles ne sont pas des globales.

11.11.2.7 Variables de requête : \$_REQUEST

Note
Introduite en 4.1.0. Dans les versions plus anciennes, elle n'avait pas d'équivalent.
Note
Dans les versions antérieures à PHP 4.3.0, les informations de \$_FILES étaient incluses également dans \$_REQUEST .
Un tableau associatif constitué du contenu des variables \$_GET , \$_POST , \$_COOKIE .

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global \$_REQUEST; pour y accéder.

Si la directive `register_globals` est active, alors ces variables seront aussi rendues directement accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux \$_REQUEST . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre [Utiliser la directive Register Globals](#) . Ces variables globales individuelles ne sont pas des globales.

11.11.2.8 Session variables : \$_SESSION

Note
Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait \$HTTP_SESSION_VARS .
Un tableau associatif des valeurs stockées dans les sessions, et accessible au script courant. Elle est automatiquement globale dans tous les contextes d'exécution. Voyez l'extension Sessions pour plus de détails sur comment est utilisée cette variable.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global \$_SESSION; pour y accéder, comme vous le faisiez avec \$HTTP_SESSION_VARS .

\$HTTP_SESSION_VARS contient les mêmes informations, mais n'est pas autoglobale. (Notez que \$HTTP_SESSION_VARS et \$_SESSION sont des variables différentes et que PHP les traite comme tel)

Si la directive `register_globals` est active, alors ces variables seront aussi rendues directement accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux \$_SESSION et \$HTTP_SESSION_VARS . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre [Utiliser la directive Register Globals](#) . Ces variables globales individuelles ne sont pas des globales.

11.11.2.9 Variables globales : \$GLOBALS

Note
\$GLOBALS est disponible depuis PHP 3.0.0.
Un tableau associatif contenant les références sur toutes les variables globales actuellement définies dans le contexte d'exécution global du script. Les noms des variables sont les index du tableau.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global \$GLOBALS; pour y accéder.

11.11.2.10 Le dernier message d'erreur : `$php_errormsg`

`$php_errormsg` est une variable qui contient le texte de la dernière erreur générée par PHP. Cette variable sera uniquement accessible dans le même contexte d'exécution que celui de la ligne qui a généré l'erreur, et uniquement si la directive de configuration `track_errors` est activée (elle est désactivée par défaut).

11.11.3 Classes pré-définies

Cette section liste les classes standards prédéfinies. Les autres extensions qui définissent d'autres classes sont décritent dans leur référence.

11.11.3.1 Classes standards

Ces classes sont définies dans le jeu de classe standard de PHP, inclus dans toutes les versions de PHP.

- Directory
La classe qui permet d'instancier `dir`.
- stdClass
- __PHP_Incomplete_Class

11.11.3.2 Classes pré-définies en PHP 5

Ces classe supplémentaires ont été ajoutées en PHP 5.0.0.

- exception
- php_user_filter

11.11.4 Constantes pré-définies

11.11.4.1 Constantes pré-définies

Ces constantes sont définies par le coeur de PHP. Cela inclut notamment PHP, le moteur Zend et les modules SAPI.

- PHP_VERSION (chaîne de caractères)
- PHP_OS (chaîne de caractères)
- PHP_EOL (chaîne de caractères)
Disponible depuis PHP 4.3.10 et PHP 5.0.2
- PHP_INT_MAX (integer)
Disponible depuis PHP 4.4.0 et PHP 5.0.5
- PHP_INT_SIZE (integer)
Disponible depuis PHP 4.4.0 et PHP 5.0.5
- DEFAULT_INCLUDE_PATH (chaîne de caractères)
- PEAR_INSTALL_DIR (chaîne de caractères)
- PEAR_EXTENSION_DIR (chaîne de caractères)
- PHP_EXTENSION_DIR (chaîne de caractères)
- PHP_BINDIR (chaîne de caractères)

PHP_LIBDIR (chaîne de caractères)
PHP_DATADIR (chaîne de caractères)
PHP_SYSCONFDIR (chaîne de caractères)
PHP_LOCALSTATEDIR (chaîne de caractères)
PHP_CONFIG_FILE_PATH (chaîne de caractères)
PHP_OUTPUT_HANDLER_START (entier)
PHP_OUTPUT_HANDLER_CONT (entier)
PHP_OUTPUT_HANDLER_END (entier)
E_ERROR (entier)
E_WARNING (entier)
E_PARSE (entier)
E_NOTICE (entier)
E_CORE_ERROR (entier)
E_CORE_WARNING (entier)
E_COMPILE_ERROR (entier)
E_COMPILE_WARNING (entier)
E_USER_ERROR (entier)
E_USER_WARNING (entier)
E_USER_NOTICE (entier)
E_ALL (entier)
E_STRICT (entier)
 Disponible depuis PHP 5.0.0.
__COMPILER_HALT_OFFSET__ (entier)
 Disponible depuis PHP 5.1.0

Voir aussi les constantes magiques .

11.11.4.2 Constantes standards pré-définies

Ces constantes sont définies par défaut dans PHP.

EXTR_OVERWRITE (entier)
EXTR_SKIP (entier)
EXTR_PREFIX_SAME (entier)
EXTR_PREFIX_ALL (entier)
EXTR_PREFIX_INVALID (entier)
EXTR_PREFIX_IF_EXISTS (entier)
EXTR_IF_EXISTS (entier)
SORT_ASC (entier)
SORT_DESC (entier)
SORT_REGULAR (entier)
SORT_NUMERIC (entier)
SORT_STRING (entier)
CASE_LOWER (entier)
CASE_UPPER (entier)
COUNT_NORMAL (entier)
COUNT_RECURSIVE (entier)
ASSERT_ACTIVE (entier)
ASSERT_CALLBACK (entier)
ASSERT_BAIL (entier)
ASSERT_WARNING (entier)
ASSERT_QUIET_EVAL (entier)
CONNECTION_ABORTED (entier)

CONNECTION_NORMAL (entier)
 CONNECTION_TIMEOUT (entier)
 INI_USER (entier)
 INI_PERDIR (entier)
 INI_SYSTEM (entier)
 INI_ALL (entier)
 M_E (float)
 M_LOG2E (float)
 M_LOG10E (float)
 M_LN2 (float)
 M_LN10 (float)
 M_PI (float)
 M_PI_2 (float)
 M_PI_4 (float)
 M_1_PI (float)
 M_2_PI (float)
 M_2_SQRTPI (float)
 M_SQRT2 (float)
 M_SQRT1_2 (float)
 CRYPT_SALT_LENGTH (entier)
 CRYPT_STD_DES (entier)
 CRYPT_EXT_DES (entier)
 CRYPT_MD5 (entier)
 CRYPT_BLOWFISH (entier)
 DIRECTORY_SEPARATOR (chaîne de caractères)
 SEEK_SET (entier)
 SEEK_CUR (entier)
 SEEK_END (entier)
 LOCK_SH (entier)
 LOCK_EX (entier)
 LOCK_UN (entier)
 LOCK_NB (entier)
 HTML_SPECIALCHARS (entier)
 HTML_ENTITIES (entier)
 ENT_COMPAT (entier)
 ENT_QUOTES (entier)
 ENT_NOQUOTES (entier)
 INFO_GENERAL (entier)
 INFO_CREDITS (entier)
 INFO_CONFIGURATION (entier)
 INFO_MODULES (entier)
 INFO_ENVIRONMENT (entier)
 INFO_VARIABLES (entier)
 INFO_LICENSE (entier)
 INFO_ALL (entier)
 CREDITS_GROUP (entier)
 CREDITS_GENERAL (entier)
 CREDITS_SAPI (entier)
 CREDITS_MODULES (entier)
 CREDITS_DOCS (entier)
 CREDITS_FULLPAGE (entier)
 CREDITS_QA (entier)
 CREDITS_ALL (entier)
 STR_PAD_LEFT (entier)

STR_PAD_RIGHT (entier)
STR_PAD_BOTH (entier)
PATHINFO_DIRNAME (entier)
PATHINFO_BASENAME (entier)
PATHINFO_EXTENSION (entier)
PATH_SEPARATOR (chaîne de caractères)
CHAR_MAX (entier)
LC_CTYPE (entier)
LC_NUMERIC (entier)
LC_TIME (entier)
LC_COLLATE (entier)
LC_MONETARY (entier)
LC_ALL (entier)
LC_MESSAGES (entier)
ABDAY_1 (entier)
ABDAY_2 (entier)
ABDAY_3 (entier)
ABDAY_4 (entier)
ABDAY_5 (entier)
ABDAY_6 (entier)
ABDAY_7 (entier)
DAY_1 (entier)
DAY_2 (entier)
DAY_3 (entier)
DAY_4 (entier)
DAY_5 (entier)
DAY_6 (entier)
DAY_7 (entier)
ABMON_1 (entier)
ABMON_2 (entier)
ABMON_3 (entier)
ABMON_4 (entier)
ABMON_5 (entier)
ABMON_6 (entier)
ABMON_7 (entier)
ABMON_8 (entier)
ABMON_9 (entier)
ABMON_10 (entier)
ABMON_11 (entier)
ABMON_12 (entier)
MON_1 (entier)
MON_2 (entier)
MON_3 (entier)
MON_4 (entier)
MON_5 (entier)
MON_6 (entier)
MON_7 (entier)
MON_8 (entier)
MON_9 (entier)
MON_10 (entier)
MON_11 (entier)
MON_12 (entier)
AM_STR (entier)
PM_STR (entier)

D_T_FMT (entier)
 D_FMT (entier)
 T_FMT (entier)
 T_FMT_AMPM (entier)
 ERA (entier)
 ERA_YEAR (entier)
 ERA_D_T_FMT (entier)
 ERA_D_FMT (entier)
 ERA_T_FMT (entier)
 ALT_DIGITS (entier)
 INT_CURR_SYMBOL (entier)
 CURRENCY_SYMBOL (entier)
 CRNCYSTR (entier)
 MON_DECIMAL_POINT (entier)
 MON_THOUSANDS_SEP (entier)
 MON_GROUPING (entier)
 POSITIVE_SIGN (entier)
 NEGATIVE_SIGN (entier)
 INT_FRAC_DIGITS (entier)
 FRAC_DIGITS (entier)
 P_CS_PRECEDES (entier)
 P_SEP_BY_SPACE (entier)
 N_CS_PRECEDES (entier)
 N_SEP_BY_SPACE (entier)
 P_SIGN_POSN (entier)
 N_SIGN_POSN (entier)
 DECIMAL_POINT (entier)
 RADIXCHAR (entier)
 THOUSANDS_SEP (entier)
 THOUSEP (entier)
 GROUPING (entier)
 YESEXPR (entier)
 NOEXPR (entier)
 YESSTR (entier)
 NOSTR (entier)
 CODESET (entier)
 LOG_EMERG (entier)
 LOG_ALERT (entier)
 LOG_CRIT (entier)
 LOG_ERR (entier)
 LOG_WARNING (entier)
 LOG_NOTICE (entier)
 LOG_INFO (entier)
 LOG_DEBUG (entier)
 LOG_KERN (entier)
 LOG_USER (entier)
 LOG_MAIL (entier)
 LOG_DAEMON (entier)
 LOG_AUTH (entier)
 LOG_SYSLOG (entier)
 LOG_LPR (entier)
 LOG_NEWS (entier)
 LOG_UUCP (entier)
 LOG_CRON (entier)

LOG_AUTHPRIV ([entier](#))
 LOG_LOCAL0 ([entier](#))
 LOG_LOCAL1 ([entier](#))
 LOG_LOCAL2 ([entier](#))
 LOG_LOCAL3 ([entier](#))
 LOG_LOCAL4 ([entier](#))
 LOG_LOCAL5 ([entier](#))
 LOG_LOCAL6 ([entier](#))
 LOG_LOCAL7 ([entier](#))
 LOG_PID ([entier](#))
 LOG_CONS ([entier](#))
 LOG_ODELAY ([entier](#))
 LOG_NDELAY ([entier](#))
 LOG_NOWAIT ([entier](#))
 LOG_PERROR ([entier](#))

11.12 Types des ressources PHP

Voici la liste des fonctions qui créent, utilisent ou détruisent les ressources PHP. Vous pouvez déterminer si une variable contient une ressource avec la fonction [is_resource](#), et le type d'une ressource que vous utilisez avec la fonction [get_resource_type](#).

Ressource	Construite par	Utilisé par	Détruite par	D
aspell	aspell_new	aspell_check , aspell_check_raw , aspell_suggest	Aucun	D A
bzip2	bzopen	bzerrno , bzerror , bzerrstr , bzflush , bzread , bzwrite	bzclose	F
COM	com_load	com_invoke , com_propget , com_get , com_propput , com_set , com_propput	Aucun	R s C
VARIANT				
cpdf	cpdf_open	cpdf_page_init , cpdf_finalize_page , cpdf_finalize , cpdf_output_buffer , cpdf_save_to_file , cpdf_set_current_page , cpdf_begin_text , cpdf_end_text , cpdf_show , cpdf_show_xy , cpdf_text , cpdf_set_font , cpdf_set_leading , cpdf_set_text_rendering , cpdf_set_horiz_scaling , cpdf_set_text_rise , cpdf_set_text_matrix , cpdf_set_text_pos , cpdf_set_text_pos , cpdf_set_word_spacing , cpdf_continue_text , cpdf_stringwidth , cpdf_save , cpdf_translate , cpdf_restore , cpdf_scale , cpdf_rotate , cpdf_setflat , cpdf_setlinejoin , cpdf_setlinecap , cpdf_setmiterlimit , cpdf_setlinewidth , cpdf_setdash , cpdf_moveto , cpdf_rmoveto , cpdf_curveto , cpdf_lineto , cpdf_rlineto	cpdf_close	D P b C

		, cpdf_circle , cpdf_arc , cpdf_rect , cpdf_closepath , cpdf_stroke , cpdf_closepath fill stroke , cpdf_fill_stroke , cpdf_clip , cpdf_fill , cpdf_setgray_fill , cpdf_setgray_stroke , cpdf_setgray , cpdf_setrgbcolor_fill , cpdf_setrgbcolor_stroke , cpdf_setrgbcolor , cpdf_add_outline , cpdf_set_page_animation , cpdf_import_jpeg , cpdf_place_inline_image , cpdf_add_annotation		
cpdf outline				
curl	curl_copy_handle , curl_init	curl_copy_handle , curl_errno , curl_error , curl_exec , curl_getinfo , curl_setopt	curl_close	S
dbm	dbmopen	dbmexists , dbmfetch , dbminsert , dbmreplace , dbmdelete , dbmfirstkey , dbmnextkey	dbmclose	L b d D
dba	dba_popen	dba_delete , dba_exists , dba_fetch , dba_firstkey , dba_insert , dba_nextkey , dba_optimize , dba_replace , dba_sync	dba_close	L b d D
dba persistent	dba_open	dba_delete , dba_exists , dba_fetch , dba_firstkey , dba_insert , dba_nextkey , dba_optimize , dba_replace , dba_sync	Aucun	L p v b d D
dbase	dbase_open	dbase_pack , dbase_add_record , dbase_replace_record , dbase_delete_record , dbase_get_record , dbase_get_record_with_names , dbase_numfields , dbase_numrecords	dbase_close	L b d d
dbx_link_object	dbx_connect	dbx_query	dbx_close	C d
dbx_result_object	dbx_query		Aucun	R
domxml attribute				
domxml document				
domxml node				
xpath context				
xpath object				
fbsql link	fbsql_change_user , fbsql_connect	fbsql_autocommit , fbsql_blob_size , fbsql_clob_size , fbsql_commit , fbsql_change_user , fbsql_create_blob , fbsql_create_db , fbsql_create_clob , fbsql_data_seek , fbsql_database_password ,	fbsql_close	L b d ft

		fbsql_database , fbsql_db_query , fbsql_db_status , fbsql_drop_db , fbsql_erro , fbsql_error , fbsql_get_autostart_info , fbsql_hostname , fbsql_insert_id , fbsql_list_dbs , fbsql_password , fbsql_read_blob , fbsql_read_clob , fbsql_rollback , fbsql_select_db , fbsql_set_password , fbsql_set_transaction , fbsql_start_db , fbsql_stop_db , fbsql_username		
fbsql plink	fbsql_change_user , fbsql_pconnect	fbsql_autocommit , fbsql_change_user , fbsql_create_db , fbsql_data_seek , fbsql_db_query , fbsql_drop_db , fbsql_select_db , fbsql_erro , fbsql_error , fbsql_insert_id , fbsql_list_dbs	Aucun	L p v b d ft
fbsql result	fbsql_db_query , fbsql_list_dbs , fbsql_query , fbsql_list_fields , fbsql_list_tables , fbsql_tablename	fbsql_affected_rows , fbsql_fetch_array , fbsql_fetch_assoc , fbsql_fetch_field , fbsql_fetch_lengths , fbsql_fetch_object , fbsql_fetch_row , fbsql_field_flags , fbsql_field_name , fbsql_field_len , fbsql_field_seek , fbsql_field_table , fbsql_field_type , fbsql_next_result , fbsql_num_fields , fbsql_num_rows , fbsql_result , fbsql_num_rows	fbsql_free_result	R ft
fdf	fdf_open	fdf_create , fdf_save , fdf_get_value , fdf_set_value , fdf_next_field_name , fdf_set_ap , fdf_set_status , fdf_get_status , fdf_set_file , fdf_get_file , fdf_set_flags , fdf_set_opt , fdf_set_submit_form_action , fdf_set_javascript_action	fdf_close	F
ftp	ftp_connect , ftp_ssl_connect	ftp_login , ftp_pwd , ftp_cdup , ftp_chdir , ftp_mkdir , ftp_rmdir , ftp_nlist , ftp_rawlist , ftp_systype , ftp_pasv , ftp_get , ftp_fget , ftp_put , ftp_fput , ftp_size , ftp_mdtm , ftp_rename , ftp_delete , ftp_site , ftp_alloc , ftp_chmod , ftp_exec , ftp_get_option , ftp_nb_continue , ftp_nb_fget , ftp_nb_fput , ftp_nb_get , ftp_nb_put , ftp_raw , ftp_set_option	ftp_close	C F
gd	imagecreate , imagecreatefromgd , imagecreatefromgd2 , imagecreatefromgd2part , imagecreatefromgif , imagecreatefromjpeg , imagecreatefrompng , imagecreatefromwbmp , imagecreatefromstring , imagecreatefromxbm ,	imagearc , imagechar , imagecharup , imagecolorallocate , imagecolorat , imagecolorclosest , imagecolorexact , imagecolorresolve , imagegammacorrect , imagegammacorrect , imagecolorset , imagecolorsforindex , imagecolorstotal , imagecolortransparent , imagecopy , imagecopyresized , imagedashedline , imagefill , imagefilledpolygon ,	imagedestroy	In

	<u>imagecreatefromxpm</u> , <u>imagecreatetruecolor</u> , <u>imagerotate</u>	<u>imagefilledrectangle</u> , <u>imagefilltoborder</u> , <u>imagegif</u> , <u>imagepng</u> , <u>imagejpeg</u> , <u>imagewbmp</u> , <u>imageinterlace</u> , <u>imageline</u> <u>, imagepolygon</u> , <u>imagepstext</u> , <u>imagerectangle</u> , <u>imagerotate</u> , <u>imagesetpixel</u> , <u>imagestring</u> , <u>imagestringup</u> , <u>imagesx</u> , <u>imagesy</u> , <u>imagefttext</u> , <u>imagefilledarc</u> , <u>imageellipse</u> , <u>imagefilledellipse</u> , <u>imagecolorclosestalpha</u> , <u>imagecolorexactalpha</u> , <u>imagecolorresolvealpha</u> , <u>imagecopymerge</u> , <u>imagecopymergegray</u> , <u>imagecopyresampled</u> , <u>imagetruecolortopalette</u> , <u>imagesetbrush</u> , <u>imagesettile</u> , <u>imagesetthickness</u> , <u>image2wbmp</u> , <u>imagealphablending</u> , <u>imageantialias</u> , <u>imagecolorallocatealpha</u> , <u>imagecolorclosesthwb</u> , <u>imagecolordeallocate</u> , <u>imagecolormatch</u> , <u>imagefilter</u> , <u>imagefttext</u> , <u>imagegd</u> , <u>imagegd2</u> , <u>imageistruecolor</u> , <u>imagelayereffect</u> , <u>imagepalettecopy</u> , <u>imagesavealpha</u> , <u>imagesetstyle</u> , <u>imagexbm</u>		
gd font	<u>imageloadfont</u>	<u>imagechar</u> , <u>imagecharup</u> , <u>imagefontheight</u>	Aucun	P G
gd PS encoding				
gd PS font	<u>imagepsloadfont</u>	<u>imagepstext</u> , <u>imagepslantfont</u> , <u>imagepsextendfont</u> , <u>imagepsencodefont</u> , <u>imagepsbbox</u>	<u>imagepsfreefont</u>	P P p
GMP integer	<u>gmp_init</u>	<u>gmp_intval</u> , <u>gmp_strval</u> , <u>gmp_add</u> , <u>gmp_sub</u> , <u>gmp_mul</u> , <u>gmp_div_q</u> , <u>gmp_div_r</u> , <u>gmp_div_qr</u> , <u>gmp_div</u> , <u>gmp_mod</u> , <u>gmp_divexact</u> , <u>gmp_cmp</u> , <u>gmp_neg</u> , <u>gmp_abs</u> , <u>gmp_sign</u> , <u>gmp_fact</u> , <u>gmp_sqrt</u> , <u>gmp_sqrtrm</u> , <u>gmp_perfect_square</u> , <u>gmp_pow</u> , <u>gmp_powm</u> , <u>gmp_prob_prime</u> , <u>gmp_gcd</u> , <u>gmp_gcdext</u> , <u>gmp_invert</u> , <u>gmp_legendre</u> , <u>gmp_jacobi</u> , <u>gmp_random</u> , <u>gmp_and</u> , <u>gmp_or</u> , <u>gmp_xor</u> , <u>gmp_setbit</u> , <u>gmp_clrbit</u> , <u>gmp_scan0</u> , <u>gmp_scan1</u> , <u>gmp_popcount</u> , <u>gmp_hamdist</u>	Aucun	N G
hyperwave document	<u>hw_cp</u> , <u>hw_docbyanchor</u> <u>, hw_getremote</u> , <u>hw_getremotechildren</u>	<u>hw_children</u> , <u>hw_childrenobj</u> , <u>hw_getparents</u> , <u>hw_getparentsobj</u> , <u>hw_getchildcoll</u> , <u>hw_getchildcollobj</u> , <u>hw_getremote</u> , <u>hw_getsrcbydestobj</u> , <u>hw_getandlock</u> , <u>hw_gettext</u> , <u>hw_getobjectbyquerycoll</u> ,	<u>hw_deleteobject</u>	C H

		hw_getobjectbyquerycollobj , hw_getchilddoccoll , hw_getchilddoccollobj , hw_getanchors , hw_getanchorsobj , hw_inscoll , hw_pipedocument , hw_unlock		
hyperwave link	hw_connect	hw_children , hw_childrenobj , hw_cp , hw_deleteobject , hw_docbyanchor , hw_docbyanchorobj , hw_errormsg , hw_edittest , hw_error , hw_getparents , hw_getparentsobj , hw_getchildcoll , hw_getchildcollobj , hw_getremote , hw_getremotechildren , hw_getsrcbydestobj , hw_getobject , hw_getandlock , hw_gettext , hw_getobjectbyquery , hw_getobjectbyqueryobj , hw_getobjectbyquerycoll , hw_getobjectbyquerycollobj , hw_getchilddoccoll , hw_getchilddoccollobj , hw_getanchors , hw_getanchorsobj , hw_mv , hw_incollections , hw_info , hw_inscoll , hw_insdoc , hw_insertdocument , hw_insertobject , hw_mapid , hw_modifyobject , hw_pipedocument , hw_unlock , hw_who , hw_getusername	hw_close , hw_free document	L s H
hyperwave link persistent	hw_pconnect	hw_children , hw_childrenobj , hw_cp , hw_deleteobject , hw_docbyanchor , hw_docbyanchorobj , hw_errormsg , hw_edittest , hw_error , hw_getparents , hw_getparentsobj , hw_getchildcoll , hw_getchildcollobj , hw_getremote , hw_getremotechildren , hw_getsrcbydestobj , hw_getobject , hw_getandlock , hw_gettext , hw_getobjectbyquery , hw_getobjectbyqueryobj , hw_getobjectbyquerycoll , hw_getobjectbyquerycollobj , hw_getchilddoccoll , hw_getchilddoccollobj , hw_getanchors , hw_getanchorsobj , hw_mv , hw_incollections , hw_info , hw_inscoll , hw_insdoc , hw_insertdocument , hw_insertobject , hw_mapid , hw_modifyobject , hw_pipedocument , hw_unlock , hw_who , hw_getusername	Aucun	L p v s H
icap	icap_open	icap_fetch_event , icap_list_events , icap_store_event , icap_snooze , icap_list_alarms , icap_delete_event	icap_close	L s
imap	imap_open	imap_append , imap_body , imap_check , imap_createmailbox , imap_delete , imap_deletemailbox ,	imap_close	L s (I

		imap_expunge , imap_fetchbody , imap_fetchstructure , imap_headerinfo , imap_header , imap_headers , imap_listmailbox , imap_getmailboxes , imap_get_quota , imap_status , imap_listsubscribed , imap_set_quota , imap_set_quota , imap_getsubscribed , imap_mail_copy , imap_mail_move , imap_num_msg , imap_num_recent , imap_ping , imap_renamemailbox , imap_reopen , imap_subscribe , imap_undelete , imap_unsubscribe , imap_scanmailbox , imap_mailboxmsginfo , imap_fetchheader , imap_uid , imap_msgno , imap_search , imap_fetch_overview		P
imap chain persistent				
imap persistent				
ingres	ingres_connect	ingres_query , ingres_num_rows , ingres_num_fields , ingres_field_name , ingres_field_type , ingres_field_nullable , ingres_field_length , ingres_field_precision , ingres_field_scale , ingres_fetch_array , ingres_fetch_row , ingres_fetch_object , ingres_rollback , ingres_commit , ingres_autocommit	ingres_close	L b d in
ingres persistent	ingres_pconnect	ingres_query , ingres_num_rows , ingres_num_fields , ingres_field_name , ingres_field_type , ingres_field_nullable , ingres_field_length , ingres_field_precision , ingres_field_scale , ingres_fetch_array , ingres_fetch_row , ingres_fetch_object , ingres_rollback , ingres_commit , ingres_autocommit	Aucun	L p v b d in
interbase blob				
interbase link	ibase_connect	ibase_query , ibase_prepare , ibase_trans	ibase_close	L b d In
interbase link persistent	ibase_pconnect	ibase_query , ibase_prepare , ibase_trans	Aucun	L p v b d In
interbase query	ibase_prepare	ibase_execute	ibase_free_query	R In
interbase result	ibase_query		ibase_free_result	

		ibase_fetch_row , ibase_fetch_object , ibase_field_info , ibase_num_fields		R I r
ibase transaction	ibase_trans	ibase_commit	ibase_rollback	T I r
java				
ldap link	ldap_connect , ldap_search	ldap_count_entries , ldap_first_attribute , ldap_first_entry , ldap_get_attributes , ldap_get_dn , ldap_get_entries , ldap_get_values , ldap_get_values_len , ldap_next_attribute , ldap_next_entry	ldap_close	L s L
ldap result	ldap_read	ldap_add , ldap_compare , ldap_bind , ldap_count_entries , ldap_delete , ldap_errno , ldap_error , ldap_first_attribute , ldap_first_entry , ldap_get_attributes , ldap_get_dn , ldap_get_entries , ldap_get_values , ldap_get_values_len , ldap_get_option , ldap_list , ldap_modify , ldap_mod_add , ldap_mod_replace , ldap_next_attribute , ldap_next_entry , ldap_mod_del , ldap_set_option , ldap_unbind	ldap_free_result	R r L
ldap result entry				
mcal	mcal_open , mcal_popen	mcal_create_calendar , mcal_rename_calendar , mcal_rename_calendar , mcal_delete_calendar , mcal_fetch_event , mcal_list_events , mcal_append_event , mcal_store_event , mcal_delete_event , mcal_list_alarms , mcal_event_init , mcal_event_set_category , mcal_event_set_title , mcal_event_set_description , mcal_event_set_start , mcal_event_set_end , mcal_event_set_alarm , mcal_event_set_class , mcal_next_recurrence , mcal_event_set_recur_none , mcal_event_set_recur_daily , mcal_event_set_recur_weekly , mcal_event_set_recur_monthly_mday , mcal_event_set_recur_monthly_wday , mcal_event_set_recur_yearly , mcal_fetch_current_stream_event , mcal_event_add_attribute , mcal_expunge	mcal_close	L s
SWFAction				
SWFBitmap				
SWFButton				
SWFDisplayItem				

SWFFill				
SWFFont				
SWFGradient				
SWFMorph				
SWFMovie				
SWFShape				
SWFSprite				
SWFText				
SWFTextField				
mnogosearch agent				
mnogosearch result				
mysql link	mysql_connect	mysql , mysql_create_db , mysql_createdb , mysql_drop_db , mysql_drop_db , mysql_select_db , mysql_select_db	mysql_close	L b d m
mysql link persistent	mysql_pconnect	mysql , mysql_create_db , mysql_createdb , mysql_drop_db , mysql_drop_db , mysql_select_db , mysql_select_db	Aucun	L p v b d m
mysql query	mysql_db_query , mysql_list_dbs , mysql_list_fields , mysql_list_tables , mysql_query	mysql , mysql_affected_rows , mysql_data_seek , mysql_dbname , mysql_fetch_array , mysql_fetch_field , mysql_fetch_object , mysql_fetch_row , mysql_field_seek , mysql_field_table , mysql_field_flags , mysql_field_len , mysql_field_name , mysql_field_type , mysql_num_fields , mysql_num_rows , mysql_numfields , mysql_numrows , mysql_result	mysql_free_result , mysql_free_result	R m
mssql link	mssql_connect	mssql_query , mssql_select_db	mssql_close	L b d M S
mssql link persistent	mssql_pconnect	mssql_query , mssql_select_db	Aucun	L p v b d M S
mssql result	mssql_query	mssql_data_seek , mssql_fetch_array , mssql_fetch_field , mssql_fetch_object , mssql_fetch_row , mssql_field_length , mssql_field_name , mssql_field_seek , mssql_field_type , mssql_num_fields , mssql_num_rows , mssql_result	mssql_free_result	R M S

mysql link	<u>mysql_connect</u>	<u>mysql_affected_rows</u> , <u>mysql_change_user</u> , <u>mysql_create_db</u> , <u>mysql_data_seek</u> , <u>mysql_db_name</u> , <u>mysql_db_query</u> , <u>mysql_drop_db</u> , <u>mysql_errno</u> , <u>mysql_error</u> , <u>mysql_insert_id</u> , <u>mysql_list_dbs</u> , <u>mysql_list_fields</u> , <u>mysql_list_tables</u> , <u>mysql_query</u> , <u>mysql_result</u> , <u>mysql_select_db</u> , <u>mysql_tablename</u> , <u>mysql_get_host_info</u> , <u>mysql_get_proto_info</u> , <u>mysql_get_server_info</u>	<u>mysql_close</u>	L b d M
mysql link persistent	<u>mysql_pconnect</u>	<u>mysql_affected_rows</u> , <u>mysql_change_user</u> , <u>mysql_create_db</u> , <u>mysql_data_seek</u> , <u>mysql_db_name</u> , <u>mysql_db_query</u> , <u>mysql_drop_db</u> , <u>mysql_errno</u> , <u>mysql_error</u> , <u>mysql_insert_id</u> , <u>mysql_list_dbs</u> , <u>mysql_list_fields</u> , <u>mysql_list_tables</u> , <u>mysql_query</u> , <u>mysql_result</u> , <u>mysql_select_db</u> , <u>mysql_tablename</u> , <u>mysql_get_host_info</u> , <u>mysql_get_proto_info</u> , <u>mysql_get_server_info</u>	Aucun	L p v b d M
mysql result	<u>mysql_db_query</u> , <u>mysql_list_dbs</u> , <u>mysql_list_fields</u> , <u>mysql_list_processes</u> , <u>mysql_list_tables</u> , <u>mysql_query</u> , <u>mysql_unbuffered_query</u>	<u>mysql_data_seek</u> , <u>mysql_db_name</u> , <u>mysql_fetch_array</u> , <u>mysql_fetch_assoc</u> , <u>mysql_fetch_field</u> , <u>mysql_fetch_lengths</u> , <u>mysql_fetch_object</u> , <u>mysql_fetch_row</u> , <u>mysql_fetch_row</u> , <u>mysql_field_flags</u> , <u>mysql_field_name</u> , <u>mysql_field_len</u> , <u>mysql_field_seek</u> , <u>mysql_field_table</u> , <u>mysql_field_type</u> , <u>mysql_num_fields</u> , <u>mysql_num_rows</u> , <u>mysql_result</u> , <u>mysql_tablename</u>	<u>mysql_free_result</u>	R M
oci8 collection				
oci8 connection	<u>ocilogon</u> , <u>ociplagon</u> , <u>ocinlogon</u>	<u>ocicommit</u> , <u>ociserverversion</u> , <u>ocinewcursor</u> , <u>ociparse</u> , <u>ocierror</u>	<u>ocilogoff</u>	L b d C
oci8 descriptor				
oci8 server				
oci8 session				
oci8 statement	<u>ocinewdescriptor</u>	<u>ocirollback</u> , <u>ocinewdescriptor</u> , <u>ocirowcount</u> , <u>ocidefinebyname</u> , <u>ocibindbyname</u> , <u>ociexecute</u> , <u>ocinumcols</u> , <u>ocireresult</u> , <u>ocifetch</u> , <u>ocifetchinto</u> , <u>ocifetchstatement</u> , <u>ocicolumnisnull</u> , <u>ocicolumnname</u> , <u>ocicolumnsize</u> , <u>ocicolumntype</u> , <u>ocistatementtype</u> , <u>ocierror</u>	<u>ocifreestatement</u>	C C
odbc link	<u>odbc_connect</u>		<u>odbc_close</u>	

		odbc_autocommit , odbc_commit , odbc_error , odbc_errormsg , odbc_exec , odbc_tables , odbc_tableprivileges , odbc_do , odbc_prepare , odbc_columns , odbc_columnprivileges , odbc_procedurecolumns , odbc_specialcolumns , odbc_rollback , odbc_setoption , odbc_gettypeinfo , odbc_primarykeys , odbc_foreignkeys , odbc_procedures , odbc_statistics		
odbc link persistent	odbc_pconnect	odbc_autocommit , odbc_commit , odbc_error , odbc_errormsg , odbc_exec , odbc_tables , odbc_tableprivileges , odbc_do , odbc_prepare , odbc_columns , odbc_columnprivileges , odbc_procedurecolumns , odbc_specialcolumns , odbc_rollback , odbc_setoption , odbc_gettypeinfo , odbc_primarykeys , odbc_foreignkeys , odbc_procedures , odbc_statistics	Aucun	
odbc result	odbc_prepare	odbc_binmode , odbc_cursor , odbc_execute , odbc_fetch_into , odbc_fetch_row , odbc_field_name , odbc_field_num , odbc_field_type , odbc_field_len , odbc_field_precision , odbc_field_scale , odbc_longreadlen , odbc_num_fields , odbc_num_rows , odbc_result , odbc_result_all , odbc_setoption	odbc_free_result	
birdstep link				
birdstep result				
OpenSSL key	openssl_get_privatekey , openssl_get_publickey	openssl_sign , openssl_seal , openssl_open , openssl_verify	openssl_free_key	
OpenSSL X.509	openssl_x509_read	openssl_x509_parse , openssl_x509_checkpurpose	openssl_x509_free	
oracle cursor	ora_open	ora_bind , ora_columnname , ora_columnsize , ora_columntype , ora_error , ora_errorcode , ora_exec , ora_fetch , ora_fetch_into , ora_getcolumn , ora_numcols , ora_numrows , ora_parse	ora_close	
oracle link	ora_logon	ora_do , ora_error , ora_errorcode , ora_rollback , ora_commitoff , ora_commiton , ora_open , ora_commit	ora_logoff	
oracle link persistent	ora_plogon	ora_do , ora_error , ora_errorcode , ora_rollback , ora_commitoff , ora_commiton , ora_open , ora_commit	Aucun	

				d o
pdf document	<u>pdf_new</u>	<u>pdf_add_bookmark</u> , <u>pdf_add_launchlink</u> , <u>pdf_add_locallink</u> , <u>pdf_add_note</u> , <u>pdf_add_pdflink</u> , <u>pdf_add_weblink</u> , <u>pdf_arc</u> , <u>pdf_attach_file</u> , <u>pdf_begin_page</u> , <u>pdf_circle</u> , <u>pdf_clip</u> , <u>pdf_closepath</u> , <u>pdf_closepath_fill_stroke</u> , <u>pdf_closepath_stroke</u> , <u>pdf_concat</u> , <u>pdf_continue_text</u> , <u>pdf_curveto</u> , <u>pdf_end_page</u> , <u>pdf_endpath</u> , <u>pdf_fill</u> , <u>pdf_fill_stroke</u> , <u>pdf_findfont</u> , <u>pdf_get_buffer</u> , <u>pdf_get_image_height</u> , <u>pdf_get_image_width</u> , <u>pdf_get_parameter</u> , <u>pdf_get_value</u> , <u>pdf_lineto</u> , <u>pdf_moveto</u> , <u>pdf_open_ccitt</u> , <u>pdf_open_file</u> , <u>pdf_open_image_file</u> , <u>pdf_place_image</u> , <u>pdf_rect</u> , <u>pdf_restore</u> , <u>pdf_rotate</u> , <u>pdf_save</u> , <u>pdf_scale</u> , <u>pdf_setdash</u> , <u>pdf_setflat</u> , <u>pdf_setfont</u> , <u>pdf_setgray</u> , <u>pdf_setgray_fill</u> , <u>pdf_setgray_stroke</u> , <u>pdf_setlinecap</u> , <u>pdf_setlinejoin</u> , <u>pdf_setlinewidth</u> , <u>pdf_setmiterlimit</u> , <u>pdf_setpolydash</u> , <u>pdf_setrgbcolor</u> , <u>pdf_setrgbcolor_fill</u> , <u>pdf_setrgbcolor_stroke</u> , <u>pdf_set_border_color</u> , <u>pdf_set_border_dash</u> , <u>pdf_set_border_style</u> , <u>pdf_set_char_spacing</u> , <u>pdf_set_duration</u> , <u>pdf_set_font</u> , <u>pdf_set_horiz_scaling</u> , <u>pdf_set_parameter</u> , <u>pdf_set_text_pos</u> , <u>pdf_set_text_rendering</u> , <u>pdf_set_value</u> , <u>pdf_set_word_spacing</u> , <u>pdf_show</u> , <u>pdf_show_boxed</u> , <u>pdf_show_xy</u> , <u>pdf_skew</u> , <u>pdf_stringwidth</u> , <u>pdf_stroke</u> , <u>pdf_translate</u> , <u>pdf_add_thumbnail</u> , <u>pdf_arcn</u> , <u>pdf_begin_pattern</u> , <u>pdf_begin_template</u> , <u>pdf_end_pattern</u> , <u>pdf_end_template</u> , <u>pdf_initgraphics</u> , <u>pdf_makespotcolor</u> , <u>pdf_set_info</u> , <u>pdf_setcolor</u> , <u>pdf_setmatrix</u> , <u>pdf_open_memory_image</u>	<u>pdf_close</u> , <u>pdf_delete</u>	D P
pdf image	<u>pdf_open_image</u> , <u>pdf_open_image_file</u> , <u>pdf_open_memory_image</u>	<u>pdf_get_image_height</u> , <u>pdf_get_image_width</u> , <u>pdf_open_ccitt</u> , <u>pdf_place_image</u>	<u>pdf_close_image</u>	Ir u P
pdf object				
pdf outline				
Objet de grande taille pgsq	<u>pg_lo_open</u>	<u>pg_lo_open</u> , <u>pg_lo_create</u> , <u>pg_lo_read</u> , <u>pg_lo_read_all</u> , <u>pg_lo_seek</u> ,	<u>pg_lo_close</u>	C g

		<u>pg_lo_tell</u> , <u>pg_lo_unlink</u> , <u>pg_lo_write</u>		P
pgsql link	<u>pg_connect</u>	<u>pg_affected_rows</u> , <u>pg_query</u> , <u>pg_send_query</u> , <u>pg_get_result</u> , <u>pg_connection_busy</u> , <u>pg_connection_reset</u> , <u>pg_connection_status</u> , <u>pg_last_error</u> , <u>pg_last_notice</u> , <u>pg_lo_create</u> , <u>pg_lo_export</u> , <u>pg_lo_import</u> , <u>pg_lo_open</u> , <u>pg_lo_unlink</u> , <u>pg_host</u> , <u>pg_port</u> , <u>pg_dbname</u> , <u>pg_options</u> , <u>pg_copy_from</u> , <u>pg_copy_to</u> , <u>pg_end_copy</u> , <u>pg_put_line</u> , <u>pg_tty</u> , <u>pg_trace</u> , <u>pg_untrace</u> , <u>pg_set_client_encoding</u> , <u>pg_client_encoding</u> , <u>pg_metadata</u> , <u>pg_convert</u> , <u>pg_insert</u> , <u>pg_select</u> , <u>pg_delete</u> , <u>pg_update</u>	<u>pg_close</u>	L b d P
pgsql link persistent	<u>pg_pconnect</u>	<u>pg_affected_rows</u> , <u>pg_query</u> , <u>pg_send_query</u> , <u>pg_get_result</u> , <u>pg_connection_busy</u> , <u>pg_connection_reset</u> , <u>pg_connection_status</u> , <u>pg_last_error</u> , <u>pg_last_notice</u> , <u>pg_lo_create</u> , <u>pg_lo_export</u> , <u>pg_lo_import</u> , <u>pg_lo_open</u> , <u>pg_lo_unlink</u> , <u>pg_host</u> , <u>pg_port</u> , <u>pg_dbname</u> , <u>pg_options</u> , <u>pg_copy_from</u> , <u>pg_copy_to</u> , <u>pg_end_copy</u> , <u>pg_put_line</u> , <u>pg_tty</u> , <u>pg_trace</u> , <u>pg_untrace</u> , <u>pg_set_client_encoding</u> , <u>pg_client_encoding</u> , <u>pg_metadata</u> , <u>pg_convert</u> , <u>pg_insert</u> , <u>pg_select</u> , <u>pg_delete</u> , <u>pg_update</u>	Aucun	L p v b d P
pgsql result	<u>pg_execute</u> , <u>pg_query</u> , <u>pg_query_params</u> , <u>pg_get_result</u>	<u>pg_fetch_array</u> , <u>pg_fetch_object</u> , <u>pg_fetch_result</u> , <u>pg_fetch_row</u> , <u>pg_field_is_null</u> , <u>pg_field_name</u> , <u>pg_field_num</u> , <u>pg_field_prtlen</u> , <u>pg_field_size</u> , <u>pg_field_type</u> , <u>pg_last_oid</u> , <u>pg_num_fields</u> , <u>pg_num_rows</u> , <u>pg_result_error</u> , <u>pg_result_status</u>	<u>pg_free_result</u>	R P
pgsql string				
printer				
printer brush				
printer font				
printer pen				
pspell	<u>pspell_new</u> , <u>pspell_new_config</u> , <u>pspell_new_personal</u>	<u>pspell_add_to_personal</u> , <u>pspell_add_to_session</u> , <u>pspell_check</u> , <u>pspell_clear_session</u> , <u>pspell_config_ignore</u> , <u>pspell_config_mode</u> , <u>pspell_config_personal</u> ,	Aucun	D P

		pspell_config_repl , pspell_config_runtogether , pspell_config_save_repl , pspell_save_wordlist , pspell_store_replacement , pspell_suggest		
pspell config	pspell_config_create	pspell_new_config	Aucun	C P
Sablotron XSLT	xslt_create	xslt_closelog , xslt_openlog , xslt_run , xslt_set_sax_handler , xslt_erno , xslt_error , xslt_fetch_result , xslt_free	xslt_free	A X
shmop	shmop_open	shmop_read , shmop_write , shmop_size , shmop_delete	shmop_close	
sockets file descriptor set	socket	accept_connect , bind , connect , listen , read , write	close	S
sockets i/o vector				
dir	opendir	readdir , rewinddir	closedir	D
Flux	fopen , tmpfile	feof , fflush , fgetc , fgetcsv , fgets , fgetss , flock , fpassthru , fputs , fwrite , fread , fseek , ftell , fstat , ftruncate , set_file_buffer , rewind	fclose	F
pipe	popen	feof , fflush , fgetc , fgetcsv , fgets , fgetss , fpassthru , fputs , fwrite , fread	pclose	P
socket	fsockopen , pfsockopen	fflush , fgetc , fgetcsv , fgets , fgetss , fpassthru , fputs , fwrite , fread	fclose	S
sybase-db link	sybase_connect	sybase_query , sybase_select_db	sybase_close	L b d S la b d
sybase-db link persistent	sybase_pconnect	sybase_query , sybase_select_db	Aucun	L p v b d S la b d
sybase-db result	sybase_query , sybase_unbuffered_query	sybase_data_seek , sybase_fetch_array , sybase_fetch_field , sybase_fetch_object , sybase_fetch_row , sybase_field_seek , sybase_num_fields , sybase_num_rows sybase_result	sybase_free_result	R S la b D
sybase-ct link	sybase_connect	sybase_affected_rows , sybase_query , sybase_select_db	sybase_close	L b d

				S la b C
sybase-ct link persistent	<u>sybase_pconnect</u>	<u>sybase_affected_rows</u> , <u>sybase_query</u> , <u>sybase_select_db</u>	Aucun	L p v b d S la b C
sybase-ct result	<u>sybase_query</u>	<u>sybase_data_seek</u> , <u>sybase_fetch_array</u> , <u>sybase_fetch_field</u> , <u>sybase_fetch_object</u> , <u>sybase_fetch_row</u> , <u>sybase_field_seek</u> , <u>sybase_num_fields</u> , <u>sybase_num_rows</u> , <u>sybase_result</u>	<u>sybase_free_result</u>	R S la b C
sysvsem	<u>sem_get</u>	<u>sem_acquire</u>	<u>sem_release</u>	S S
sysvshm	<u>shm_attach</u>	<u>shm_remove</u> , <u>shm_put_var</u> , <u>shm_get_var</u> , <u>shm_remove_var</u>	<u>shm_detach</u>	M p S
wddx	<u>wddx_packet_start</u>	<u>wddx_add_vars</u>	<u>wddx_packet_end</u>	P
xml	<u>xml_parser_create</u> , <u>xml_parser_create_ns</u>	<u>xml_set_object</u> , <u>xml_set_element_handler</u> , <u>xml_set_character_data_handler</u> , <u>xml_set_processing_instruction_handler</u> , <u>xml_set_default_handler</u> , <u>xml_set_unparsed_entity_decl_handler</u> , <u>xml_set_notation_decl_handler</u> , <u>xml_set_external_entity_ref_handler</u> , <u>xml_parse</u> , <u>xml_get_error_code</u> , <u>xml_error_string</u> , <u>xml_get_current_line_number</u> , <u>xml_get_current_column_number</u> , <u>xml_get_current_byte_index</u> , <u>xml_parse_into_struct</u> , <u>xml_parser_set_option</u> , <u>xml_parser_get_option</u>	<u>xml_parser_free</u>	A s X
zlib	<u>gzopen</u>	<u>gzeof</u> , <u>gzgetc</u> , <u>gzgets</u> , <u>gzgetss</u> , <u>gzpassthru</u> , <u>gzputs</u> , <u>gzread</u> , <u>gzrewind</u> , <u>gzseek</u> , <u>gztell</u> , <u>gzwrite</u>	<u>gzclose</u>	F c G

11.13 Liste des protocoles supportés

Cette section recense une liste de protocoles gérant les URL, qui sont intégrés dans les fonctions d'accès aux fichiers, dans PHP. Par exemple, les fonctions fopen et copy . Ces fonctionnalités sont compilées comme des gestionnaires externes, et à partir de PHP 4.3.0, vous pouvez créer vos propres gestionnaires, avec la fonction stream_register_wrapper .

11.13.1 Système de fichiers

Toutes les versions de PHP. Explicitement avec le protocole file:// depuis PHP 4.3.0.

- /path/to/file.ext
- relative/path/to/file.ext
- fileInCwd.ext
- C:/path/to/winfile.ext
- C:\path\to\winfile.ext
- \\smbserver\share\path\to\winfile.ext
- file:///path/to/file.ext

file:// est le gestionnaire par défaut de PHP et il représente les fichiers locaux. Lorsqu'un chemin relatif est spécifié (un chemin qui ne commence pas par /, \, \\, ou une lettre de lecteur Windows), le chemin sera calculé relativement à la position courante. Dans de nombreux cas, c'est le dossier de résidence du script, à moins qu'il n'ait été modifié. En utilisant la version CLI, le chemin sera calculé par rapport au dossier d'appel du script.

Avec certaines fonctions comme `fopen` et `file_get_contents`, `include_path` peut être scanné pour y trouver les fichiers, si un chemin relatif est fourni.

Attribut	Supporté
Restreint par <code>allow_url_fopen</code> .	Non
Autorise les lectures	Oui
Autorise les écritures	Oui
Autorise l'ajout	Oui
Autorise simultanément les lectures et écritures	Oui
Supporte <code>stat</code>	Oui
Supporte <code>unlink</code>	Oui
Supporte <code>rename</code>	Oui
Supporte <code>mkdir</code>	Oui
Supporte <code>rmdir</code>	Oui

11.13.2 Socket

Cette section contient les options de contexte supportées par les gestionnaires qui travaillent via les sockets, comme tcp, http ou ftp.

Depuis PHP 5.1.0, une seule option est supportée, `bindto`, qui peut être utilisée pour spécifier l'adresse IP (soit IPv4 ou IPv6) et/ou le numéro du port que PHP utilisera pour accéder au réseau. La syntaxe est ip:port (vous pouvez définir l'IP ou le numéro du port à 0 si vous voulez laisser le choix au système).

Note
Sachant que FTP crée deux sockets pendant une opération normale, vous ne pouvez pas spécifier le numéro du port dans l'option <code>bindto</code> . Donc, la seule syntaxe supportée est ip:0 pour le gestionnaire FTP.
Quelques exemples sur l'utilisation de l'option <code>bindto</code>
<pre><?php // Connexion à Internet en utilisant l'IP '192.168.0.100' \$options = array('socket' => array('bindto' => '192.168.0.100:0'));</pre>

```
// Connexion à Internet en utilisant l'IP '192.168.0.100' et le port '7000'
$opts = array('socket' =>
array('bindto' => '192.168.0.100:7000'));

// Connexion à Internet en utilisant le port '7000'
$opts = array('socket' =>
array('bindto' => '0:7000'));

// Création du contexte...
$context = stream_context_create($opts);

// ...et on l'utilise pour récupérer les données
echo file_get_contents('http://www.example.com', false, $context);

?>
```

11.13.3 HTTP et HTTPS

PHP 3, PHP 4, PHP 5. https:// depuis PHP 4.3.0

- http://example.com
- http://example.com/file.php?var1=val1&var2=val2
- http://user:motdepasse@example.com
- https://example.com
- https://example.com/file.php?var1=val1&var2=val2
- https://user:motdepasse@example.com

Permet des accès en lecture uniquement, pour des fichiers accessibles sur le réseau avec la méthode GET de HTTP 1.0. Un en-tête Host: est envoyé avec la requête, pour gérer les hôtes virtuels, basés sur des noms. Si vous avez configuré une version de navigateur avec l'option user_agent dans votre fichier php.ini , ou via le contexte de flux, il sera aussi utilisé dans votre requête.

Attention

Lorsque vous utilisez SSL, le serveur IIS de Microsoft violera le protocole en fermant la connexion sans envoyer l'indicateur `close_notify`. PHP le reportera en tant que "SSL: Fatal Protocol Error" quand vous arrivez à la fin des données. L'astuce est de baisser le niveau de la directive error_reporting pour ne pas inclure les warnings. A partir de PHP 4.3.7, le bogue est détecté automatiquement lors de l'ouverture du flux en utilisant `https://` et supprimera cet avertissement pour vous. Si vous utilisez fsockopen pour créer une socket `ssl://`, vous devez vous occuper vous-même de supprimer l'erreur.

Les redirections sont supportées depuis PHP 4.0.5; si vous utilisez une version plus ancienne, vous devrez ajouter vous-même les slash finaux dans vos URL. S'il est important de connaître l'URL de la ressource qui vous a fourni le document, une fois que toutes les redirections ont été suivies, vous devez analyser les en-têtes de réponses retournés par le flux.

http:// et redirections

```
<?php
$url = 'http://www.example.com/redirecting_page.php';

$fp = fopen($url, 'r');

/* Avant PHP 4.3.0 utilisez $http_response_header
```

```

    au lieu de stream_get_meta_data() */
$meta_data = stream_get_meta_data($fp);
foreach($meta_data['wrapper_data'] as $response) {

    /* Redirections? */
    if (substr(strtolower($response), 0, 18) == 'location: ') {
        /* modification de $url avec la nouvelle URL */
        $url = substr($response, 18);
    }
}
?>

```

Le flux donne l'accès au corps (**body**) de la ressource. Les en-têtes sont stockés dans la variable \$http_response_header . Depuis PHP 4.3.0, les en-têtes sont aussi accessibles avec la fonction stream_get_meta_data .

Les connexions HTTP sont en lecture seule. Vous ne pouvez pas écrire dans des fichiers ouverts via le protocole HTTP.

Note
HTTPS est supporté depuis PHP 4.3, si vous avez compilé le support d'OpenSSL.

Attribut	Supporté
Limité par <u>allow_url_fopen</u> .	Oui
Autorisé en lecture	Oui
Autorisé en écriture	Non
Autorisé en ajout	Non
Autorise les lectures / écritures simultanées	Non applicable
Supporte <u>stat</u>	Non
Supporte <u>unlink</u>	Non
Supporte <u>rename</u>	Non
Supporte <u>mkdir</u>	Non
Supporte <u>rmdir</u>	Non

Nom	Utilisation	Par défaut
method	GET , POST , ou une autre méthode HTTP acceptée apr le serveur distant.	GET
header	Les autres en-têtes envoyés durant la requête. Les valeurs de cette option remplaceront les valeurs déjà configurées (comme User-agent: , Host: , et Authentication:).	
user_agent	Valeur à envoyer pour l'en-tête User-Agent: . Cette valeur sera utilisée si un type de navigateur n'est pas spécifié dans l'option de contexte header ci-dessus.	Directive dans le php.ini : user_agent .
content	D'autres données envoyées après les en-têtes. Typiquement, utilisé pour placer les requêtes POST ou PUT.	
proxy	URI du serveur de proxy. (i.e. tcp://proxy.example.com:5100). Les proxys HTTPS (via les proxys HTTP) ne fonctionnent qu'avec PHP 5.1.0 ou supérieur.	
max_redirects	Le nombre maximum de redirection à suivre. Ajouté en PHP 5.1.0.	20

request_fulluri	Lorsque cette option vaut TRUE , l'URI entière sera utilisée lors de la construction de la requête. (i.e. GET http://www.example.com/path/to/file.html HTTP/1.0). Même si ce n'est pas la méthode standard, certains proxies le requiert.	FALSE
Note		
Options de contexte pour les sockets		
Plusieurs options de contexte supplémentaires sont supportées par le <u>gestionnaire de transport</u> Pour les flux http:// , voyez les options de contexte du transport tcp:// . Pour les flux https:// , voyez les options de contexte du transport ssl:// .		

11.13.4 FTP et FTPS

PHP 3, PHP 4, PHP 5. ftps:// depuis PHP 4.3

- ftp://example.com/pub/file.txt
- ftp://user:motdepasse@example.com/pub/file.txt
- ftps://example.com/pub/file.txt
- ftps://user:motdepasse@example.com/pub/file.txt

Permet l'accès aux fichiers existants, et la création de fichiers via FTP. Si le serveur ne supporte pas les connexions en mode passif, la connexion échouera.

Vous pouvez ouvrir des fichiers en lecture et en écriture, mais pas les deux en même temps. Si le fichier distant existe déjà sur le serveur ftp et que vous tentez de l'ouvrir en écriture mais que vous n'avez pas spécifié l'option overwrite dans le contexte, la connexion échouera. Si vous devez réécrire des fichiers existants en utilisant ftp, spécifiez l'option overwrite dans le contexte et ouvrez le fichier en écriture. Alternativement, vous pouvez utiliser l' extension FTP .

Note
Apposer
Depuis PHP 5.0.0, les fichiers doivent être apposé via le gestion d'URL ftp:// . Dans les versions plus anciennes, tenter d'apposer un fichier via ftp:// causera une erreur.
ftps:// a été introduit en PHP 4.3. C'est le même que ftp:// , mais il tente d'établir une connexion sécurisée avec le serveur FTP. Si le serveur ne le supporte pas, la connexion sera une connexion non sécurisée.

Note
FTPS est supporté depuis PHP 4.3.0, si vous avez compilé le support d'OpenSSL.

Attribut	PHP 4	PHP 5
Limité par <u>allow_url_fopen</u> .	Oui	Oui
Autorisé en lecture	Oui	Oui
Autorisé en écriture	Oui (nouveaux fichiers uniquement)	Oui (nouveaux fichiers et fichiers existants avec overwrite)
Autorisé en ajout	Non	Oui
Autorisé en lecture et écriture	Non	Non
Supporte <u>stat</u>	Non	

		Depuis PHP 5.0.0 : uniquement <code>filesize</code> , <code>filetype</code> , <code>file_exists</code> , <code>is_file</code> et <code>is_dir</code> . Depuis PHP 5.1.0 : <code>filemtime</code> .
Supporte <code>unlink</code>	Non	Oui
Supporte <code>rename</code>	Non	Oui
Supporte <code>mkdir</code>	Non	Oui
Supporte <code>rmdir</code>	Non	Oui

Nam	Utilisation	Par défaut
<code>overwrite</code>	Permet l'écrasement d'un fichier existant sur le serveur distant. S'applique au mode écriture uniquement.	FALSE (Désactivé)
<code>resume_pos</code>	Offset du fichier où commencer le transfert. S'applique uniquement au mode de lecture.	0 (Début du fichier)
<code>proxy</code> (PHP 5.1.0 ou supérieur)	Requête FTP proxy via le serveur http proxy. Appliquer uniquement aux opérations de lecture de fichiers. Exemple : <code>tcp://squid.example.com:8000</code>	
Note		
Options de contexte pour les sockets		
Plusieurs options de contexte supplémentaires sont supportées par le gestionnaire de transport Pour les flux <code>ftp://</code> , voyez les options de contexte du transport <code>tcp://</code> . Pour les flux <code>https://</code> , voyez les options de contexte du transport <code>ssl://</code> .		

11.13.5 Flux d'entrée/sortie

PHP 3.0.13 et plus récent, `php://output` et `php://input` depuis PHP 4.3

- `php://stdin`
- `php://stdout`
- `php://stderr`
- `php://output`
- `php://input`

`php://stdin` , `php://stdout` et `php://stderr` donne accès au flux correspondant d'entrée ou de sortie du processus PHP.

`php://output` vous permet d'écrire dans le buffer de sortie, de la même manière que `print` et `echo` .

`php://input` permet de lire des données POST bruts. C'est moins gourmand en mémoire que `$HTTP_RAW_POST_DATA` et il n'y a pas de directive spéciale dans `php.ini` . `php://input` n'est pas disponible avec `enctype="multipart/form-data"` .

`php://stdin` et `php://input` sont lecture seule, tandis que `php://stdout` , `php://stderr` et `php://output` sont en écriture seule.

`php://filter` est une sorte de méta-gestionnaire, prévu pour qui permet l'utilisation de filtre avec les données d'entrée au moment du démarrage du script. C'est pratique avec des fonctions compactes comme `readfile` , `file` et `file_get_contents` où il n'y a pas d'opportunité d'appliquer un filtre aux données lues.

Le gestionnaire de `php://filter` prend les 'paramètres' suivants dans le 'chemin'.

- /resource=<flux à filtrer> (**obligatoire**) Ce paramètre doit être situé à la fin de votre chaîne php://filter et doit pointer sur le flux à filtrer.

php:// et les filtres

```
<?php
/* Ceci est l'équivalent de :
   readfile("http://www.example.com");
   puisque aucun filtre n'est spécifié */

readfile("php://filter/resource=http://www.example.com");
?>
```

- /read=<liste de filtres de la chaîne de lecture> (**optionnel**) Ce paramètre prend un ou plusieurs noms de filtres, séparés par le caractère pipe | .

php:// et filtres de lecture

```
<?php
/* Cette ligne va afficher le contenu de
   www.example.com tout en majuscule */
readfile("php://filter/read=string.toupper/resource=http://www.example.com");

/* Cela va faire la même chose que ci-dessus, mais en plus,
   le résultat sera chiffré en ROT13 */
readfile("php://filter/read=string.toupper|string.rot13/resource=http://www.example.com");
?>
```

- /write=<liste de filtres de la chaîne d'écriture> (**optionnel**) Ce paramètre prend un ou plusieurs noms de filtres, séparés par le caractère pipe | .

php:// et filtres d'écritures

```
<?php
/* Cette ligne va filtrer la chaîne "Bonjour le monde"
   avec le filtre rot13, puis écrire le résultat dans
   le fichier exemple.txt du dossier courant */
file_put_contents("php://filter/write=string.rot13/resource=exemple.txt","Bonjour le monde");
?>
```

- /<liste de filtres des deux chaînes> (**optionnel**) Tous les filtres qui ne sont pas préfixé avec read= ou write= seront appliqué aux deux chaînes de filtres, en lecture et en écriture.

Attribut	Supporté
Limité par <u>allow_url_fopen</u> .	Non
Autorisé en lecture	php://stdin et php://input uniquement.
Autorisé en écriture	php://stdout , php://stderr et php://output uniquement.
Autorisé en ajout	php://stdout , php://stderr et php://output uniquement. (Equivalent à écrire)
Autorisé en lecture et écriture	Non. Ces gestionnaires sont unidirectionnels.
Supporte <u>stat</u>	Non
Supporte <u>unlink</u>	Non
Supporte <u>rename</u>	Non
Supporte <u>mkdir</u>	Non
Supporte <u>rmdir</u>	Non

11.13.6 Flux compressés

zlib: PHP 4.0.4 - PHP 4.2.3 (systèmes avec fopencookie uniquement)

compress.zlib:// et compress.bzip2:// depuis PHP 4.3.0 et plus récent.

- zlib:
- compress.zlib://
- compress.bzip2://

zlib: fonctionne comme [gzopen](#) , hormis le fait que le flux peut être utilisé directement avec [fread](#) et les autres fonctions de système de fichier. Cette notation est obsolète depuis PHP 4.3.0, étant données les ambiguïtés dues aux noms de fichiers contenant des deux points ':'. Utilisez plutôt `compress.zlib://` .

`compress.zlib://` et `compress.bzip2://` sont équivalents à [gzopen](#) et [bzopen](#) respectivement, et ils opèrent même sur les systèmes qui ne supportent pas `fopencookie`.

Attribut	Supporté
Limité par allow_url_fopen .	Non
Autorisé en lecture	Oui
Autorisé en écriture	Oui
Autorisé en ajout	Oui
Autorisé en lecture et écriture	Non
Supporte stat	Non, utilisez le gestionnaire <code>file://</code> pour avoir des informations sur les fichiers compressés.
Supporte unlink	Non, utilisez le gestionnaire <code>file://</code> pour avoir des informations sur les fichiers compressés.
Supporte rename	Non
Supporte mkdir	Non
Supporte rmdir	Non

11.13.7 Shell sécurisé 2

`ssh2.shell://` `ssh2.exec://` `ssh2.tunnel://` `ssh2.sftp://` `ssh2.scp://` PHP 4.3.0 et supérieur (PECL)

- `ssh2.shell://user:pass@example.com:22/xterm`
- `ssh2.exec://user:pass@example.com:22/usr/local/bin/somecmd`
- `ssh2.tunnel://user:pass@example.com:22/192.168.0.1:14`
- `ssh2.sftp://user:pass@example.com:22/path/to/filename`

Note
Ce gestionnaire n'est pas activé par défaut
Pour utiliser le gestionnaire <code>ssh2.*://</code> , vous devez installer l'extension SSH2 disponible dans PECL .

En plus d'accepter les traditionnelles identifications via l'URI, le gestionnaire `ssh2` réutilisera les connexions ouvertes en passant la ressource de connexion dans la partie hôte de l'URL.

Ouverture d'un flux depuis une connexion active
<pre><?php \$session = ssh2_connect('example.com', 22); ssh2_auth_pubkey_file(\$session, 'username', '/home/username/.ssh/id_rsa.pub', '/home/username/.ssh/id_rsa', 'secret');</pre>

```
$stream = fopen("ssh2.tunnel://$session/remote.example.com:1234", 'r');
?>
```

Attribut	ssh2.shell	ssh2.exec	ssh2.tunnel	ssh2.sftp	ssh2.scp
Restreint par <u>allow_url_fopen</u> .	Oui	Oui	Oui	Oui	Oui
Permet la lecture	Oui	Oui	Oui	Oui	Oui
Permet l'écriture	Oui	Oui	Oui	Oui	Non
Permet l'ajout	Non	Non	Non	Oui (lorsque supporté par le serveur)	Non
Permet l'écriture et la lecture simultanément	Oui	Oui	Oui	Oui	Non
Supporte la fonction <u>stat</u>	Non	Non	Non	Oui	Non
Supporte la fonction <u>unlink</u>	Non	Non	Non	Oui	Non
Supporte la fonction <u>rename</u>	Non	Non	Non	Oui	Non
Supporte la fonction <u>mkdir</u>	Non	Non	Non	Oui	Non
Supporte la fonction <u>rmdir</u>	Non	Non	Non	Oui	Non

Nom	Usage	Défaut	
session	ressource ssh2 pré-connectée pour être réutilisée		
sftp	ressource sftp pré-allouée pour être réutilisée		
methods	méthodes échange de clés, hostkey, cipher, compression et MAC à utiliser		
callbacks			
username	Nom de l'utilisateur pour la connexion		
password	Mot de passe à utiliser lors d'une identification via mot de passe		
pubkey_file	Nom du fichier contenant la clé publique à utiliser lors de l'identification		
privkey_file	Nom du fichier contenant la clé privée à utiliser lors de l'identification		
env	Tableau associatif de variables d'environnements à définir		
term	Type d'émulation de terminal à demander lors de l'allocation d'un pty		
term_width	Largeur du terminal à demander lors de l'allocation d'un pty		
term_height	Hauteur du terminal à demander lors de l'allocation d'un pty		
term_units	Unités à utiliser avec term_width et term_height		SSH2_TERM_UNIT_CHARS

11.13.8 Flux Audio

ogg:// PHP 4.3.0 et suivant (PECL)

- ogg://fichierSon.ogg
- ogg:///chemin/vers/fichierSon.ogg
- ogg://http://www.example.com/chemin/vers/FluxAduio.ogg

Note
Ce gestionnaire n'est pas activé par défaut
Pour utiliser le gestionnaire ogg:// , vous devez installer l'extension <u>OGG/Vorbis</u> disponible sur <u>PECL</u> .

Les fichiers ouverts pour lecture avec le gestionnaire ogg:// sont traités comme de l'audio compressé en utilisant le codec OGG/Vorbis . De la même façon, les fichiers ouverts pour écriture ou pour ajout avec le gestionnaire ogg:// sont écrits comme étant des données audio compressées. La fonction stream_get_meta_data , lorsqu'elle est utilisée sur un fichier OGG/Vorbis ouvert pour lecture retournera des nombreux détails concernant ce flux, incluant l'en-tête vendor , tous les commentaires comments , le nombre de canaux channels , le taux d'échantillonnage rate ainsi que le taux d'encodage décrit par : bitrate_lower , bitrate_upper , bitrate_nominal et bitrate_window .

Attribut	Supporté
Restreint par <u>allow_url_fopen</u>	Non
Permet la lecture	Oui
Permet l'écriture	Oui
Permet l'ajout	Oui
Permet la lecture et l'écriture simultanément	Non
Supporte la fonction <u>stat</u>	Non
Supporte la fonction <u>unlink</u>	Non
Supporte la fonction <u>rename</u>	Non
Supporte la fonction <u>mkdir</u>	Non
Supporte la fonction <u>rmdir</u>	Non

Nom	Utilisation	Défaut	Mode
pcm_mode	L'encodage PCM à appliquer pendant la lecture, parmi un de ceux là : OGGVORBIS_PCM_U8 , OGGVORBIS_PCM_S8 , OGGVORBIS_PCM_U16_BE , OGGVORBIS_PCM_S16_BE , OGGVORBIS_PCM_U16_LE et OGGVORBIS_PCM_S16_LE . (8 vs 16 bit, signé ou non, gros ou petit endian)	OGGVORBIS_PCM_S16_LE	Lecture
rate	Taux d'échantillonnage des données entrantes, exprimé en Hz	44100	Écriture/Traitement
bitrate	Lorsque donné sous forme d'entier, la vitesse du débit pour encoder. (16000 to 131072) Lorsque donné sous forme de nombre à virgule flottante, la variation de la qualité du débit à utiliser. (-1.0 to 1.0)	128000	Écriture/Ajout
channels	Le numéro du canal audio à encoder, typiquement 1 (Mono) ou 2 (Stéréo). Intervalle pouvant aller aussi haut que 16.	2	Écriture/Ajout
comments	Un tableau de valeurs sous forme de chaînes de caractères à encoder		Écriture/Ajout

_____ dans la piste d'en-tête. _____

11.13.9 Gestionnaire de flux

expect:// PHP 4.3.0 et supérieur (PECL)

- expect://command

Note
Ce gestionnaire est disponible par défaut
Pour pouvoir utiliser le gestionnaire expect:// , vous devez installer l'extension <u>Expect</u> disponible depuis <u>PECL</u> .
Les flux ouverts avec le gestionnaire expect:// permettent d'accéder à stdio, stdout et stderr via PTY.

Attribut	Supporté
Restreint par <u>allow_url_fopen</u> .	Non
Permet la lectureAllows Reading	Oui
Permet l'écriture	Oui
Permet l'ajout	Oui
Permet l'écriture et la lecture simultanément	Non
Support de la fonction <u>stat</u>	Non
Support de la fonction <u>unlink</u>	Non
Support de la fonction <u>rename</u>	Non
Support de la fonction <u>mkdir</u>	Non
Support de la fonction <u>rmdir</u>	Non

11.14 Liste des filtres standards

Cette section contient la liste des filtres de flux, à utiliser avec stream_filter_append . Votre version de PHP peut avoir des filtres supplémentaires, ou manquants par rapport à cette liste.

Il est bon de noter une légère différence entre stream_filter_append et stream_filter_prepend . Tous les flux PHP disposent d'un petit **buffer de lecture** , où ils stockent des blocs de données lues dans le système de fichiers, ou dans d'autres sources, afin de les traiter plus efficacement. Aussitôt que les données sont lues depuis la ressource dans le buffer interne, les données sont immédiatement passées au filtre, même si l'application PHP n'est pas prête pour lire ces données. Si des données sont déjà en attente dans le buffer lorsqu'un filtre est ajouté au flux, ces données seront immédiatement passées au filtre, pour que la modification soit transparente. Mais si vous ajoutez un filtre avec prepend, les données **NE seront PAS** filtrées. Elles attendront que le prochain bloc arrive de la ressource.

Pour avoir la liste complète des filtres de votre version de PHP, utilisez la fonction stream_get_filters .

11.14.1 Filtres de chaînes de caractères

Chaque filtre fait ce que son nom implique, et correspond au comportement de la fonction PHP

correspondante. Pour plus de détails sur un filtre, reportez-vous au manuel de la fonction correspondante.

`string.rot13` (depuis PHP 4.3.0) Utilisez ce filtre pour faire traiter les données par la fonction `str_rot13`.

Exemple avec le filtre `string.rot13`

```
<?php
$fp = fopen('php://output', 'w');
stream_filter_append($fp, 'string.rot13');
fwrite($fp, "Ceci est un test.\n");
/* affiche : Prpv rfg ha grfg. */
?>
```

`string.toupper` (depuis PHP 5.0.0) Utilisez ce filtre pour faire traiter les données par la fonction `strtoupper`.

Exemple avec le filtre `string.toupper`

```
<?php
$fp = fopen('php://output', 'w');
stream_filter_append($fp, 'string.toupper');
fwrite($fp, "Ceci est un test.\n");
/* affiche : CECI EST UN TEST. */
?>
```

`string.tolower` (depuis PHP 5.0.0) Utilisez ce filtre pour faire traiter les données par la fonction `strtolower`.

Exemple avec le filtre `string.tolower`

```
<?php
$fp = fopen('php://output', 'w');
stream_filter_append($fp, 'string.tolower');
fwrite($fp, "Ceci est un test.\n");
/* affiche : ceci est un test. */
?>
```

`string.strip_tags` (depuis PHP 5.0.0) Utilisez ce filtre pour faire traiter les données par la fonction `strip_tags`. Le filtre accepte les paramètres sous deux formats : soit une chaîne contenant une liste de balises, similairement au second paramètre de la fonction `strip_tags` ; soit un tableau de noms de balises.

Exemple avec le filtre `string.strip_tags`

```
<?php
$fp = fopen('php://output', 'w');
stream_filter_append($fp, 'string.strip_tags', STREAM_FILTER_WRITE, "<strong><em><span>");
fwrite($fp, "<strong>Ce texte en gras</strong> finit par être agrandi <h1>en un titre 1</h1>\n");
fclose($fp);
/* affiche : <strong>Ce texte en gras</strong> finit par être agrandi en un titre 1 */

$fp = fopen('php://output', 'w');
stream_filter_append($fp, 'string.strip_tags', STREAM_FILTER_WRITE, array('strong','em','span'));
fwrite($fp, "<strong>Ce texte en gras</strong> finit par être agrandi <h1>en un titre 1</h1>\n");
fclose($fp);
/* affiche : <strong>Ce texte en gras</strong> finit par être agrandi en un titre 1 */
?>
```

11.14.2 Filtres de conversion

Comme pour les filtres de chaînes, les filtres de conversion font ce que leur nom indique. Ces filtres ont été ajoutés en PHP 5.0.0. Pour plus d'informations, voyez le manuel de la fonction correspondante.

`convert.base64-encode` et `convert.base64-decode` Utilisez ce filtre pour faire traiter les données par les fonctions `base64_encode` et `base64_decode`. `convert.base64-encode` supporte les paramètres sous forme de tableau. Si `line-length` est donné, le résultat sera coupé en lignes de `line-length` caractères. Si `line-break-chars` est fourni, chaque bloc de caractères sera terminé par ce paramètre. Ces paramètres donnent le même effet que la fonction `base64_encode` utilisée conjointement avec `chunk_split`.

Exemple avec les filtres `convert.base64-encode` et `convert.base64-decode`

```
<?php
$fp = fopen('php://output', 'w');
stream_filter_append($fp, 'convert.base64-encode');
fwrite($fp, "This is a test.\n");
fclose($fp);
/* affiche : VGhpcyBpcyBhIHRlc3QuCg== */

$params = array('line-length' => 8, 'line-break-chars' => "\r\n");
$fp = fopen('php://output', 'w');
stream_filter_append($fp, 'convert.base64-encode', STREAM_FILTER_WRITE, $params);
fwrite($fp, "This is a test.\n");
fclose($fp);
/* affiche : VGhpcyBp
           : cyBhIHRl
           : c3QuCg== */

$fp = fopen('php://output', 'w');
stream_filter_append($fp, 'convert.base64-decode');
fwrite($fp, "VGhpcyBpcyBhIHRlc3QuCg==");
fclose($fp);
/* affiche : This is a test. */
?>
```

`convert.quoted-printable-encode` et `convert.quoted-printable-decode` Utilisez ce filtre pour faire traiter les données par la fonction `quoted_printable_encode`. Il n'y a pas d'équivalent fonctionnel à `convert.quoted-printable-encode`. `convert.quoted-printable-encode` reçoit les paramètres sous forme de tableau. En plus des paramètres classiques de `convert.base64-encode`, `convert.quoted-printable-encode` accepte aussi les arguments booléens `binary` et `force-encode-first`. `convert.base64-decode` ne supporte le paramètre `line-break-chars` que comme indication pour supprimer les nouvelles lignes.

Exemple avec les filtres `convert.quoted-printable-encode` et `convert.quoted-printable-decode`

```
<?php
$fp = fopen('php://output', 'w');
stream_filter_append($fp, 'convert.quoted-printable-encode');
fwrite($fp, "This is a test.\n");
/* affiche : =This is a test.=0A */
?>
```

11.14.3 Filtres de compression

Alors que les gestionnaires de compression fournissent un moyen pour générer des fichiers aux formats gzip et bz2, ils ne permettent pas de gérer les protocoles réseau compressés, ni de commencer avec un flux non compressé pour le transformer en flux compressé. Pour faire cela, il

faut appliquer un filtre de compression.

Note

Les filtres de compression **ne génère pas** les en-têtes et fin de fichiers, comme le fait l'utilitaire `gzip`. Ils ne font que compresser et décompresser des portions de flux de données.

`zlib.deflate` (compression) et `zlib.inflate` (décompression) sont les implémentations des méthodes de compression présentées dans la [RFC 1951](#). Le filtre `deflate` prend jusqu'à trois paramètres, passés sous la forme de tableau associatif.

`level` spécifie le niveau de compression souhaité, de 1 à 9. Plus le niveau est haut, meilleure est la compression, et plus cher est le coût de compression. Deux niveaux de compression spéciaux existent : 0, qui représente l'absence de compression, et -1, qui représente le niveau par défaut de la bibliothèque `zlib` : actuellement, c'est 6.

`window` est la taille du buffer mémoire, en base 2. Les valeurs supérieures, jusqu'à 15, soit 32768 octets, donnent de meilleures compressions, et les valeurs inférieures, jusqu'à 9, soit 512 octets, occupent le moins d'espace en mémoire. Par défaut, `window` vaut actuellement 15.

`memory` est une indication du niveau de mémoire nécessaire. Les valeurs valides vont de 1, pour l'allocation minimale, à 9, pour une allocation maximale. L'allocation de mémoire affecte la vitesse d'exécution, et non pas le coût global.

Note

Comme le niveau de compression est le paramètre le plus courant, il peut aussi être fourni en passant un entier comme paramètre, au lieu d'un tableau.

Les filtres de compression `zlib.*` sont disponibles avec PHP depuis la version 5.1.0 si le support `zlib` est activé. Ils sont également disponibles pour les versions 5.0.x en installant le paquet `zlib_filter` depuis [PECL](#). Ces filtres ne sont **pas** disponibles en PHP 4.

zlib.deflate et zlib.inflate

```
<?php
$params = array('level' => 6, 'window' => 15, 'memory' => 9);

$original_text = "This is a test.\nThis is only a test.\nThis is not an important string.\n";
echo "Le texte original est long de " . strlen($original_text) . " octets.\n";

$fp = fopen('test.deflated', 'w');
stream_filter_append($fp, 'zlib.deflate', STREAM_FILTER_WRITE, $params);
fwrite($fp, $original_text);
fclose($fp);

echo "Le fichier compressé fait " . filesize('test.deflated') . " octets de long.\n";
echo "Le texte original était :\n";
/* Utilise readfile et zlib.inflate pour décompresser à la volée */
readfile('php://filter/zlib.inflate/resource=test.deflated');

/* Affiche :

Le texte original est long de 70 octets
Le fichier compressé fait 56 octets de long.
Le texte original était :
This is a test.
This is only a test.
This is not an important string.

*/
?>
```

simple zlib.deflate

```

<?php
$original_text = "This is a test.\nThis is only a test.\nThis is not an important string.\n";
echo "Le texte original est long de " . strlen($original_text) . " octets.\n";

$fp = fopen('test.deflated', 'w');
/* Ici, "6" indique le niveau de compression de 6 */
stream_filter_append($fp, 'zlib.deflate', STREAM_FILTER_WRITE, 6);
fwrite($fp, $original_text);
fclose($fp);

echo "Le fichier compressé fait " . filesize('test.deflated') . " octets de long.\n";

/* Affiche :

Le texte original est long de 70 octets
Le fichier compressé fait 56 octets de long.

*/
?>

```

bzip2.compress et bzip2.decompress fonctionnent de la même manière que les filtres zlib.

Le filtre bzip2.compress accepte jusqu'à 2 paramètres, sous la forme d'un tableau associatif :

blocks est une valeur entière, de 1 à 9, an integer value from 1 to 9 specifying the number of 100kbyte blocks of memory to allocate for workspace.

work est aussi un entier dont la valeur va de 0 à 250, et qui indique le niveau d'efforts fournis avec une méthode de compression avant de passer à une autre méthode, plus lente. Modifier ce paramètre n'a qu'un effet sur la vitesse de compression. Le gain d'espace ou la mémoire utilisée restent les mêmes. Un niveau de 0 indique que la bibliothèque doit utiliser sa configuration par défaut.

Le filtre bzip2.decompress accepte uniquement un paramètre, qui peut être passé sous la forme d'un booléen associé à l'index small .

small , lorsqu'il vaut TRUE , indique à la bibliothèque bzip qu'elle doit faire une décompression en utilisant le moins de mémoire possible, aux dépens de la vitesse.

Les filtres bzip2.* ne sont pas inclus dans la distribution PHP depuis la version 5.1.0 si le support [bz2](#) est activé. Ils sont également disponibles pour les versions 5.0.x de PHP en installant le paquet [bz2_filter](#) depuis [PECL](#) . Ces filtres ne sont **pas** disponibles pour PHP 4.

bzip2.compress et bzip2.decompress

```

<?php
$param = array('blocks' => 9, 'work' => 0);

echo "Le texte original est long de " . strlen(LICENSE) . " octets.\n";

$fp = fopen('LICENSE.compressed', 'w');
stream_filter_append($fp, 'bzip2.compress', STREAM_FILTER_WRITE, $param);
fwrite($fp, file_get_contents('LICENSE'));
fclose($fp);

echo "Le fichier compressé fait " . filesize('LICENSE.compressed') . " octets de long.\n";

/* Affiche :

Le texte original est long de 3288 octets.

```

Le fichier compressé fait 1488 octets de long.

```
*/
?>
```

11.14.4 Filtres de chiffrement

mcrypt.* et mdecrypt.* fournissent un système de chiffrement symétrique basé sur libmcrypt. Les deux filtres supportent les mêmes algorithmes disponibles avec la bibliothèque `mcrypt`, sous la forme de la directive `mcrypt.ciphername` où `ciphername` est le nom du chiffrement qui sera passé à `mcrypt_module_open`. Les cinq paramètres suivants sont aussi disponibles :

Paramètre	Obligatoire	Par défaut	Valeurs possibles
mode	Optionnel	cbc	cbc, cfb, ecb, nofb, ofb, stream
algorithms_dir	Optionnel	<code>ini_get('mcrypt.algorithms_dir')</code>	Chemin vers les algorithmes de modules
modes_dir	Optionnel	<code>ini_get('mcrypt.modes_dir')</code>	Chemin vers les modes de modules
iv	Obligatoire	N/A	Généralement 8, 16 ou 32 octets de données binaires. Dépend du chiffrement
key	Obligatoire	N/A	Généralement 8, 16 ou 32 octets de données binaires. Dépend du chiffrement

Chiffrement d'un fichier avec 3DES

```
<?php
$passphrase = 'My secret';

/* Transforme un mot de passe humain
 * en une paire iv/clé
 */
$iv = substr(md5('iv'.$passphrase, true), 0, 8);
$key = substr(md5('pass1'.$passphrase, true) .
              md5('pass2'.$passphrase, true), 0, 24);
$options = array('iv'=>$iv, 'key'=>$key);

$fp = fopen('secert-file.enc', 'wb');
stream_filter_append($fp, 'mcrypt.tripledes', STREAM_FILTER_WRITE, $options);
fwrite($fp, 'Secret secret secret data');
fclose($fp);
?>
```

Lecture d'un fichier chiffré

```
<?php
$passphrase = 'My secret';

/* Transforme un mot de passe humain
 * en une paire iv/clé
 */
$iv = substr(md5('iv'.$passphrase, true), 0, 8);
$key = substr(md5('pass1'.$passphrase, true) .
              md5('pass2'.$passphrase, true), 0, 24);
$options = array('iv'=>$iv, 'key'=>$key);

$fp = fopen('secert-file.enc', 'rb');
stream_filter_append($fp, 'mdecrypt.tripledes', STREAM_FILTER_WRITE, $options);
$data = rtrim(stream_get_contents($fp));
fclose($fp);

echo $data;
?>
```

11.15 Liste des modes de transports de sockets disponibles

Voici la liste des différents modes de transports, format URL, dont PHP dispose en interne pour les flux qui exploitent les sockets, tels que `fsocketopen` et `stream_socket_client`. Ces modes de transports **ne s'appliquent pas** à l'extension `sockets`.

Pour connaître la liste des modes de transports installés sur votre version de PHP, utilisez `stream_get_transports`.

11.15.1 Domaines Internet : TCP, UDP, SSL et TLS

PHP 3, PHP 4, PHP 5. `ssl://` & `tls://` depuis PHP 4.30, `sslv2://` & `sslv3://` depuis PHP 5.0.2.

Note

Si aucun transport n'est spécifié, <code>tcp://</code> est utilisé.

- 127.0.0.1
- fe80::1
- www.example.com
- tcp://127.0.0.1
- tcp://fe80::1
- tcp://www.example.com
- udp://www.example.com
- ssl://www.example.com
- sslv2://www.example.com
- sslv3://www.example.com
- tls://www.example.com

Les sockets du domaine Internet utilisent un numéro de port en plus de l'adresse de l'hôte. Dans le cas de `fsocketopen`, il est spécifié en deuxième paramètre, et donc, n'a pas d'impact sur le format du mode de transport. Avec `stream_socket_client` et les autres fonctions de la même famille, le numéro de port est spécifié comme un suffixe dans l'URL de transport, identifié par un signe deux-points.

- tcp://127.0.0.1:80
- tcp://[fe80::1]:80
- tcp://www.example.com:80

Note

Adresse IPv6 et numéro de port

Dans le second exemple ci-dessus, les exemples en IPv4 et les noms d'hôtes sont identiques, mais les IPv6 sont placées entre crochet, en plus d'avoir le deux-points et le numéro de port : <code>[fe80::1]</code> . Cela permet de distinguer les deux-points utilisés en IPv6 et le deux-points utilisé pour délimiter le numéro de port.

Les modes `ssl://` et `tls://` (disponibles uniquement lorsque le support OpenSSL est compilé avec PHP) sont des extensions de `tcp://` qui incluent le chiffrement SSL. En PHP 4.3, OpenSSL doit être compilé statiquement avec PHP. en PHP 5.0, il peut être compilé en module ou statiquement.

`ssl://` va tenter de négocier une connexion SSL V2 ou SSL V3, suivant les capacités et les références de l'hôte distant. `sslv2://` et `sslv3://` sélectionnent explicitement le protocole.

Nom	Utilisation	Valeur par défaut
verify_peer	TRUE ou FALSE . Requiert la vérification du certificat SSL utilisé.	FALSE
allow_self_signed	TRUE ou FALSE . Permet les certificats auto-signés.	FALSE
cafile	Situation du fichier de l'autorité de certification sur le serveur local, qui doit être utilisé avec verify_peer comme option de contexte pour identifier un pair distant.	
capath	Si cafile n'est pas spécifié, ou si le certificat n'est pas trouvé, le dossier capath est scanné pour y trouver un certificat convenable. capath doit être un dossier de certificats corrects.	
local_cert	Chemin jusqu'au fichier de certificat local. Il doit être au format PEM, qui contient votre certificat, et la clé privée. Il peut éventuellement contenir la chaîne de certificat de l'émetteur.	
passphrase	Mot de passe avec lequel votre local_cert a été chiffré.	
CN_match	Le nom commun (Common Name) que nous attendons. PHP va effectuer une comparaison avec jokers limités. Si le nom commun ne correspond pas à celui attendu, la connexion va échouer.	
Note		
Comme ssl:// est le mode de transport sous-jacent de https:// et ftps:// , toutes les options de contexte qui s'applique à ssl:// , s'appliquent aussi à https:// et ftps:// .		

11.15.2 Domaine Unix : UNIX et UDG

unix:// depuis PHP 3, udg:// depuis PHP 5

- unix:///tmp/mysock
- udg:///tmp/mysock

unix:// fournit l'accès à un flux de type socket, sur un domaine Unix. udg:// fournit un mode de transport alternatif, avec un protocole de datagrammes utilisateurs.

Les sockets du domaine Unix, contrairement à celle du domaine Internet, n'utilisent pas de numéro de port. Dans ce cas, le paramètre portno de [fsockopen](#) doit valoir 0.

11.16 Comparaison de types en PHP

La table suivante résume les différents comportement de PHP avec les [types](#) et [opérateurs de comparaison](#) , strictes ou larges. Cette table est aussi reliée au [transtypage](#) . Elle a été inspirée par différents commentaires d'utilisateurs, et par le travail fait chez [BlueShoes](#) .

Avant d'utiliser ces tables, il est important de comprendre les types et leur signification. Par exemple, "42" est une [chaîne de caractères](#) , alors que 42 est un [entier](#) . FALSE est boolean alors que "false" est une [chaîne de caractères](#) .

Note
Les formulaires HTML ne connaissent pas les entiers, nombres à virgules et autres booléens. Pour savoir si une structure est un entier, utilisez is_numeric .
Note

La ligne if (\$x) génère une erreur de niveau E_NOTICE lorsque \$x est indéfini. Alternativement, utilisez plutôt les fonctions `empty` ou `isset`, ou encore, initialisez toutes vos variables.

Expression	<code>gettype</code>	<code>empty</code>	<code>is_null</code>	<code>isset</code>	boolean : if(\$x)
<code>\$x = "";</code>	chaîne de caractères	TRUE	FALSE	TRUE	FALSE
<code>\$x = NULL</code>	NULL	TRUE	TRUE	FALSE	FALSE
<code>var \$x;</code>	NULL	TRUE	TRUE	FALSE	FALSE
<code>\$x est indéfini</code>	NULL	TRUE	TRUE	FALSE	FALSE
<code>\$x = array();</code>	array	TRUE	FALSE	TRUE	FALSE
<code>\$x = false;</code>	boolean	TRUE	FALSE	TRUE	FALSE
<code>\$x = true;</code>	boolean	FALSE	FALSE	TRUE	TRUE
<code>\$x = 1;</code>	entier	FALSE	FALSE	TRUE	TRUE
<code>\$x = 42;</code>	entier	FALSE	FALSE	TRUE	TRUE
<code>\$x = 0;</code>	entier	TRUE	FALSE	TRUE	FALSE
<code>\$x = -1;</code>	entier	FALSE	FALSE	TRUE	TRUE
<code>\$x = "1";</code>	chaîne de caractères	FALSE	FALSE	TRUE	TRUE
<code>\$x = "0";</code>	chaîne de caractères	TRUE	FALSE	TRUE	FALSE
<code>\$x = "-1";</code>	chaîne de caractères	FALSE	FALSE	TRUE	TRUE
<code>\$x = "php";</code>	chaîne de caractères	FALSE	FALSE	TRUE	TRUE
<code>\$x = "true";</code>	chaîne de caractères	FALSE	FALSE	TRUE	TRUE
<code>\$x = "false";</code>	chaîne de caractères	FALSE	FALSE	TRUE	TRUE

	TRUE	FALSE	1	0	-1	"1"	"0"	"-1"	NULL	array()	"php"
TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE
FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE
1	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
0	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE
-1	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
"1"	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
"0"	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
"-1"	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
NULL	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE
array()	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE
"php"	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE

	TRUE	FALSE	1	0	-1	"1"	"0"	"-1"	NULL	array()	"php"
TRUE	TRUE	FALSE	FALSE								
FALSE	FALSE	TRUE	FALSE	FALSE							
1	FALSE	FALSE	TRUE	FALSE	FALSE						
0	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
-1	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
"1"	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
"0"	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
"-1"	FALSE	TRUE	FALSE	FALSE	FALSE						
NULL	FALSE	TRUE	FALSE	FALSE							

array()	FALSE	TRUE	FALSE								
"php"	FALSE	TRUE									
Note											
Note PHP 3.0											
La chaîne de valeur "0" était considéré comme non vide en PHP 3. Ce comportement a changé en PHP 4, où il est désormais considéré comme vide.											

11.17 Liste des token de l'analyseur

De nombreuses parties du langage PHP sont représentées en interne par des tokens. Littéralement, ce sont des trucs, ou jetons. Ici, des briques de base, comme T_SR. PHP affiche des identifiants comme celui-ci dans les erreurs d'analyse, comme : unexpected T_SR, expecting ',' or ';' in script.php on line 10. .

Vous êtes supposés savoir ce que T_SR signifie. Pour ceux qui ne le savent pas, voici une liste de ces identifiant, la syntaxe PHP correspondante, et les références appropriées au manuel.

Token	Syntaxe	Référence
T_ABSTRACT	abstract	Abstraction de classes (disponible depuis PHP 5.0.0)
T_AND_EQUAL	&=	opérateurs d'assignation
T_ARRAY	array()	array , syntaxe de tableau
T_ARRAY_CAST	(array)	transtypage
T_AS	as	foreach
T_BAD_CHARACTER		Tous les caractères en dessous de ASCII 32 excepté \t (0x09), \n (0x0a) et \r (0x0d)
T_BOOLEAN_AND	&&	opérateurs logiques
T_BOOLEAN_OR		opérateurs logiques
T_BOOL_CAST	(bool) ou (boolean)	transtypage
T_BREAK	break;	break
T_CASE	case	switch
T_CATCH	catch	Exceptions (disponible depuis PHP 5.0.0)
T_CHARACTER		
T_CLASS	class	classes et objets
T_CLONE	clone	classes et objets . PHP 5 uniquement.
T_CLOSE_TAG	?> ou %>	
T_COMMENT	// ou #, et /* */ en PHP 5	commentaires
T_CONCAT_EQUAL	.=	opérateurs d'assignation
T_CONST	const	
T_CONSTANT_ENCAPSED_STRING	"foo" ou 'bar'	syntaxe chaîne de caractères
T_CONTINUE	continue	
T_CURLY_OPEN		
T_DEC	--	opérateurs d'incrémention/décrémention

T_DECLARE	declare	declare
T_DEFAULT	default	switch
T_DIV_EQUAL	/=	opérateurs d'assignation
T_DNUMBER	0.12, etc.	nombres à virgule flottante
T_DOC_COMMENT	/** */	style de commentaire dans la PHPDoc (PHP 5 uniquement)
T_DO	do	do...while
T_DOLLAR_OPEN_CURLY_BRACES	\${	syntaxe de variable complexe analysée
T_DOUBLE_ARROW	=>	syntaxe de tableau
T_DOUBLE_CAST	(real), (double) ou (float)	transtypage
T_DOUBLE_COLON	::	Voyez T_PAAMAYIM_NEKUDOTAYIM plus bas
T_ECHO	echo	echo
T_ELSE	else	else
T_ELSEIF	elseif	elseif
T_EMPTY	empty	empty
T_ENCAPSED_AND_WHITESPACE		
T_ENDDECLARE	enddeclare	declare, syntaxe alternative
T_ENDFOR	endfor	for, syntaxe alternative
T_ENDFOREACH	endforeach	foreach, syntaxe alternative
T_ENDIF	endif	if, syntaxe alternative
T_ENDSWITCH	endswitch	switch, syntaxe alternative
T_ENDWHILE	endwhile	while, syntaxe alternative
T_END_HEREDOC		syntaxe heredoc
T_EVAL	eval()	eval
T_EXIT	exit or die	exit , die
T_EXTENDS	extends	extends, classes et objets
T_FILE	__FILE__	constantes
T_FINAL	final	Mot clé "final" (disponible depuis PHP 5.0.0)
T_FOR	for	for
T_FOREACH	foreach	foreach
T_FUNCTION	function or cfunction	fonctions
T_GLOBAL	global	scope de variable
T_HALT_COMPILER	__halt_compiler()	__halt_compiler (disponible depuis PHP 5.1.0)
T_IF	if	if
T_IMPLEMENTS	implements	Interfaces (disponible depuis PHP 5.0.0)
T_INC	++	opérateurs d'incrémention/décrémention
T_INCLUDE	include()	include
T_INCLUDE_ONCE	include_once()	include_once
T_INLINE_HTML		
T_INSTANCEOF	instanceof	opérateurs de type (PHP5 seulement)
T_INT_CAST	(int) ou (integer)	transtypage

T_INTERFACE	interface	Interfaces (dipsonible depuis PHP 5.0.0)
T_ISSET	isset()	isset
T_IS_EQUAL	==	opérateurs de comparaison
T_IS_GREATER_OR_EQUAL	>=	opérateurs de comparaison
T_IS_IDENTICAL	===	opérateurs de comparaison
T_IS_NOT_EQUAL	!= ou <>	opérateurs de comparaison
T_IS_NOT_IDENTICAL	!==	opérateurs de comparaison
T_IS_SMALLER_OR_EQUAL	<=	opérateurs de comparaison
T_LINE	__LINE__	constants
T_LIST	list()	list
T_LNUMBER	123, 012, 0x1ac, etc	entiers
T_LOGICAL_AND	and	opérateurs logiques
T_LOGICAL_OR	or	opérateurs logiques
T_LOGICAL_XOR	xor	opérateurs logiques
T_MINUS_EQUAL	-=	opérateurs d'assignation
T_ML_COMMENT	/* et */	commentaires (PHP 4 uniquement)
T_MOD_EQUAL	%=	opérateurs d'assignation
T_MUL_EQUAL	*=	opérateurs d'assignation
T_NEW	new	classes et objets
T_NUM_STRING		
T_OBJECT_CAST	(object)	transtypage
T_OBJECT_OPERATOR	->	classes et objets
T_OLD_FUNCTION	old function	old_function
T_OPEN_TAG	<?php, <? or <%	sortie du mode HTML
T_OPEN_TAG_WITH_ECHO	<?=? ou <%=	sortie du mode HTML
T_OR_EQUAL	=	opérateurs d'assignation
T_PAAMAYIM_NEKUDOTAYIM	::	:: . Définie également en tant que T_DOUBLE_COLON .
T_PLUS_EQUAL	+=	opérateurs d'assignation
T_PRINT	print()	print
T_PRIVATE	private	classes et objets . PHP 5 uniquement.
T_PUBLIC	public	classes et objets . PHP 5 uniquement.
T_PROTECTED	protected	classes et objets . PHP 5 uniquement.
T_REQUIRE	require()	require
T_REQUIRE_ONCE	require_once()	require_once
T_RETURN	return	valeurs retournées
T_SL	<<	opérateurs sur les bits
T_SL_EQUAL	<<=	opérateurs d'assignation
T_SR	>>	opérateurs sur les bits
T_SR_EQUAL	>>=	opérateurs d'assignation
T_START_HEREDOC	<<<	syntaxe heredoc
T_STATIC	static	scope de variable
T_STRING		
T_STRING_CAST	(string)	transtypage

T_STRING_VARNAME		
T_SWITCH	switch	switch
T_THROW	throw	Exceptions (disponible depuis PHP 5.0.0)
T_TRY	try	Exceptions (disponible depuis PHP 5.0.0)
T_UNSET	unset()	unset
T_UNSET_CAST	(unset)	(non documenté; forcé à NULL)
T_USE	use	(pas implémenté)
T_VAR	var	classes et objets
T_VARIABLE	\$foo	variables
T_WHILE	while	while, do...while
T_WHITESPACE		
T_XOR_EQUAL	^=	opérateurs d'assignation
T_FUNC_C	__FUNCTION__	constantes , depuis PHP 4.3.0
T_CLASS_C	__CLASS__	constantes , depuis PHP 4.3.0

Voir aussi [token_name](#) .

11.18 A propos du manuel

11.18.1 Formats

Le manuel PHP est fourni en différents formats. Ces formats sont divisés en deux groupes : ceux qui sont disponibles en ligne, et les téléchargeables.

Note
Certains éditeurs ont fournis des versions imprimées de ce manuel. Nous n'en recommandons aucune, car elles sont rapidement obsolètes.

Le manuel peut être lu en ligne sur le site php.net et sur les nombreux miroirs. Pour un confort d'utilisation accru, il est recommandé d'utiliser un miroir proche de chez vous. La version HTML du manuel PHP est soit en texte simple (et donc, facilement imprimable), ou bien aux couleurs de php.net.

Deux avantages du manuel en ligne sur la plupart des formats téléchargeables, est l'intégration des [notes des utilisateurs](#) et les [URL de raccourci](#) que vous pouvez utiliser pour accéder rapidement à une partie du manuel. Un inconvénient évident est que vous devez être en ligne pour profiter de ce format.

Il y a de nombreux formats de manuel pour la consultation hors ligne, et le format le plus approprié dépend de votre OS et de vos goûts personnels. Pour savoir comment le manuel est généré, lisez la section '[Comment le manuel est généré](#)' de cet appendice.

Le format le plus portable est le HTML. Le manuel est fourni dans un format en une seule page HTML, ou comme un ensemble de fichiers de tailles réduites (mais un bon millier de fichiers tout de même). Nous fournissons ce format sous une forme compressée, vous aurez donc besoin d'un utilitaire de décompression pour extraire les fichiers de l'archive.

Pour les plates-formes Windows, le format Windows HTML Help fournit une version HTML du manuel à utiliser avec l'application Windows HTML Help : il intègre un moteur de recherche complet, un index et des signets. De nombreux IDE sous Windows fournissent des liens avec ce format pour une meilleure intégration. Il existe aussi des visualisateurs de CHM pour Linux. Visitez [xCHM](#) ou [GnoCHM](#) .

Il existe aussi une [version CHM étendue](#) , qui est mise à jour moins souvent mais qui fournit plus de fonctionnalités. Elle ne fonctionnera que sur Microsoft Windows à cause des technologies utilisées pour construire ces pages.

11.18.2 A propos des notes utilisateurs

Les notes des utilisateurs jouent un rôle très important dans le développement de ce manuel. En permettant aux lecteurs de contribuer par des exemples, des remarques et critiques, ou encore des clarifications, nous intégrons des aspects très importants du langage dans le manuel. Jusqu'à ce que les notes les plus importantes soient intégrées dans la documentation, elles sont disponibles sur le site lui-même, et dans certains formats hors ligne.

Note
Les notes des utilisateurs ne sont pas modérées avant d'apparaître sur le site, et même si elles sont modérées, leur véracité ne peut être garantie, pas plus qu'il n'existe de garantie quand à l'exactitude du manuel lui-même.
Note
Pour des questions de portée de licence, les notes utilisateurs sont considérées comme faisant partie du manuel PHP, et sont donc couvertes par la même licence qui couvre cette documentation (Open Publication License à ce jour). Pour plus de détails, lisez la page Manual's Copyright .

11.18.3 Comment lire la définition d'une fonction (prototype)

Chaque fonction dans le manuel est documentée pour permettre une compréhension rapide. Savoir décoder le texte rendra votre apprentissage plus facile. Plutôt que de dépendre d'exemples prêts en copier/coller, il est plus utile de savoir lire la définition d'une fonction (prototype). Voici comment :

Note
Pré-requis : Connaissances de base des types .
Bien que PHP soit un langage sans typage fort, une connaissance de base des types est essentielle, car ils ont quand même des sens importants.

Les définitions de fonctions vous indiquent quel type de données est [retourné](#) . Examinons la fonction [strlen](#) comme exemple :

`strlen`

(PHP 3, PHP 4, PHP 5)

`strlen --` Retourne la taille de la chaîne

Description

`int strlen (string str)`

`strlen()` retourne la taille de la chaîne `str`.

Partie	Description
strlen	Le nom de la fonction.
(PHP 3, PHP 4, PHP 5)	strlen() est présente dans toutes les versions de PHP 3, PHP 4 et PHP 5.
int	Type de valeur retourné par cette fonction, qui est en l'occurrence un <u>entier</u> (i.e. la taille d'une chaîne est mesurée par un nombre).
(string str)	Le premier (et ici le seul) paramètre à fournir à cette fonction est le paramètre str , qui doit être du type <u>chaîne de caractères</u> .

Nous pourrions réécrire ce prototype avec une version plus générique :

```
type_retourné nom_de_la_fonction ( type_du_paramètre nom_du_paramètre )
```

Plusieurs fonctions ont besoin de plusieurs paramètres, comme in_array . Son prototype est le suivant :

```
bool in_array ( mixed needle, array haystack [, bool strict])
```

Qu'est ce que cela signifie? in_array retourne un booléen TRUE en cas de réussite (le paramètre needle a été trouvé dans le tableau haystack) ou FALSE en cas d'échec (le paramètre needle n'a pas été trouvé dans le tableau haystack). Le premier paramètre s'appelle needle et il peut être de différents types : il porte donc la mention **mixed** . Le paramètre needle (ce que nous recherchons) peut être une valeur scalaire (chaîne de caractères , entier , ou float), ou encore un array . haystack (le tableau dans lequel nous recherchons) est le second paramètre. Le troisième paramètre, strict , est optionnel. Tous les paramètres optionnels sont placés entre crochets. La manuel indique que le paramètre strict vaut par défaut FALSE . Reportez-vous au manuel de chaque fonction pour savoir comment elle fonctionne.

Il y'a aussi des fonctions avec des informations plus complexes concernant les versions de PHP. Prenons html_entity_decode comme exemple :

```
(PHP 4 >= 4.3.0, PHP 5)
```

Cela signifie que cette fonction n'est pas disponible en PHP 3 et qu'elle n'est disponible que depuis PHP 4.3.0.

11.18.4 Versions de PHP documentées dans ce manuel

Cette documentation contient des informations sur PHP 4 et PHP 5, avec des notes compatibilités et de migration depuis PHP 3. Le comportement, les paramètres, les valeurs retournées, etc. qui ont changé entre deux versions du manuel PHP sont documentées dans les notes utilisateurs, et dans le manuel lui-même.

Des portions de la documentation existent pour la version CVS de PHP, ce qui signifie que les toutes dernières versions de développement de PHP proposent ces fonctionnalités. Ces versions n'étant pas encore sorties, elles ne sont disponibles que via le système de gestion des versions CVS ou via un snapshot . Soyez donc prudent avec les fonctionnalités marquées comme "Uniquement disponible sur le CVS" car elles sont normalement pas disponibles. Ces fonctionnalités seront plutôt disponibles dans la prochaine version stable de PHP. Pour télécharger la version CVS, lisez la page d'accès à CVS .

De plus, la documentation peut être disponible pour une version de PHP qui n'est pas encore publiée (par exemple, des fonctions destinées à la version PHP 5.0.0 tandis que la version publique actuelle est PHP 4.3.x). La plupart du temps, ce n'est pas une erreur de documentation, mais une indication sur les fonctionnalités qui seront disponibles avec la prochaine publication de PHP. Typiquement, PHP n'ajoute de nouvelles fonctionnalités que dans les nouvelles versions, sinon, seuls les bogues sont corrigés. En utilisant le format de version A.B.C, une version majeure incrémente A ou B tandis que une version mineure incrémente C. Donc, par exemple, il n'est pas marginal pour une fonctionnalité d'être documentée comme étant disponible en PHP x.1.x alors que la dernière version est PHP x.0.x. Notez également que le manuel est écrit dans le présent, et non dans le futur.

A plusieurs reprises, le manuel de PHP liste les valeurs par défaut des directives PHP. Ces valeurs sont basées sur le fichier `php.ini-dist` et non sur le fichier `php.ini-recommended`. Elles font également références à la dernière version de PHP. Lisez [l'annexe sur les directives PHP](#) pour plus de détails sur ces valeurs et leurs modifications.

11.18.5 Où trouver plus d'informations sur PHP ?

Ce manuel n'a pas pour objectif de fournir des présentations sur les pratiques de programmation. Si vous êtes un néophyte total, ou même un programmeur débutant, vous pouvez trouver difficile d'apprendre la programmation PHP avec uniquement ce manuel : il serait mieux de trouver des ressources plus orientées vers l'apprentissage. Vous pouvez aussi trouver une liste des livres traitant de PHP sur <http://www.php.net/books.php>.

Il y a un bon nombre de listes de diffusion actives, traitant de tous les aspects du langage et de la programmation PHP. Si vous êtes bloqués par un problème, vous pourrez sûrement trouver de l'aide auprès de ces listes. Il existe un recensement des listes de diffusion sur [la page de support de php.net](#). De plus, sur [la page des liens de php.net](#), il y'a une liste de sites web dédiés à PHP, avec des articles, des forums et des bibliothèques de code.

11.18.6 Comment aider à l'amélioration de la documentation

Il y a trois façon de participer à l'amélioration de la documentation.

Si vous trouvez des erreurs, dans n'importe quelle traduction de la documentation, utilisez le système de rapport de bogues : <http://bugs.php.net/>. Classez le rapport sous la catégorie "Documentation Problem". Vous pouvez aussi poster les problèmes liés aux différents formats du manuel à cet endroit.

Note
N'abusez pas du système de bogues pour envoyer des demandes d'aide. Utilisez plutôt une des options proposées par le support .

En contribuant aux notes, vous pouvez fournir de nouveaux exemples, mettre en lumière des effets de bords et apporter des clarifications pour les autres lecteurs. Mais ne prenez pas le système d'annotation pour un système de soumission de bogues. Vous pouvez en savoir plus sur les annotations dans la section ['A propos des notes utilisateurs'](#)

Le manuel PHP est traduit en de nombreux langages. La connaissance de l'anglais ainsi que d'une autre langue peut vous permettre d'aider la documentation de PHP en travaillant avec une équipe de traduction. Pour plus d'informations sur le commencement d'une traduction, ou sur l'aide d'une version déjà traduite, commencez par lire <http://doc.php.net/php/dochowto/>.

Le projet de documentation de PHP a un canal IRC où vous pouvez venir et parler avec les auteurs du manuel ou trouver un certain aspect du manuel avec lequel vous pourriez aider. Les coordonnées : #phpdoc sur irc.freenode.net .

11.18.7 Comment sont générées les documentations

Ce manuel est écrit en XML , en utilisant [la DTD de DocBook XML](#) , et [DSSSL](#) (Document Style and Semantics Specification Language) pour le format, et aussi, expérimentalement, [XSLT](#) (eXtensible Stylesheet Language Transformations) pour les maintenances et les formats.

Grâce au format XML comme source, nous avons la possibilité de générer de nombreux formats tout en ayant une seule source pour tous les formats. Les outils utilisés pour formater les versions HTML et TEX sont [Jade](#) , écrit par [James Clark](#) et [The Modular DocBook Stylesheets](#) écrit par [Norman Walsh](#) . Nous utilisons Array pour générer le format Windows HTML Help du manuel, et bien sur, PHP lui-même pour réaliser certaines conversions et formats.

Le manuel PHP est généré en de nombreux langages et formats, lisez <http://www.php.net/docs.php> pour plus d'informations. Le code source XML peut être téléchargé depuis le CVS et lû sur <http://cvs.php.net/> . La documentation est stocké dans le module phpdoc .

11.18.8 Traductions

La manuel PHP est disponible non seulement en anglais, mais aussi dans différentes langues. Le texte du manuel est écrit d'abord en anglais, puis des équipes à travers le monde, assure la traduction du manuel dans leur langue natale. Si la traduction d'une section n'est pas encore disponible, le système de création de la documentation présentera alors la version anglaise.

Les contributeurs aux documentations partent des codes sources XML disponibles sur le CVS <http://cvs.php.net/> puis traduisent dans leur langue. Ils **n'utilisent pas** les versions générées (comme le HTML ou le texte plein) car c'est le système d'édition qui se charge de faire les conversions du format XML vers un format lisible.

Note

Si vous voulez aider à la traduction de la documentation, entrez en contact avec l'équipe de documentation en vous inscrivant à la liste de diffusion : phpdoc-subscribe@lists.php.net . L'adresse de la liste de diffusion est phpdoc@lists.php.net . Indiquez dans le message que vous êtes intéressé par la traduction de la documentation dans une nouvelle langue, et quelqu'un viendra vous aider à démarrer une nouvelle traduction, ou rejoindre l'équipe qui a pris en charge cette traduction.

Actuellement, le manuel est disponible, partiellement ou en totalité, dans les langues suivantes : Allemand, Chinois (simplifié), Chinois (traditionnel), Chinois (contonais de Hong Kong), Coréen, Espagnol, Finnois, Français, Grec, Hébreu, Hongrois, Italien, Japonais, Néerlandais, Polonais, Portugais brésilien, Roumain, Russe, Slovaque, Slovène, Suédois, Tchèque et Turc.

Ils peuvent tous être téléchargés ici : <http://www.php.net/docs.php> .

11.18.9 Traduction française

La version française est traduite quotidiennement et disponible chez Nexen (nexen.net/). Ce manuel a été généré à partir de la documentation originale en anglais du PHP Documentation

Group, au format XML. Elle est 100% compatible avec le système de génération du groupe PHP. Elle est traité avec un système développé par Nexen pour produire divers autres formats.

Pour les problèmes liés à la traduction, vous pouvez contacter directement les éditeurs de ce manuel sur doc-fr@lists.php.net .

11.19 Open Publication License

La documentation PHP est publiée sous licence Open Publication License . Voici le texte de la licence.

v1.0, 8 June 1999

11.19.1 I. REQUIREMENTS ON BOTH UNMODIFIED AND MODIFIED VERSIONS

The Open Publication works may be reproduced and distributed in whole or in part, in any medium physical or electronic, provided that the terms of this license are adhered to, and that this license or an incorporation of it by reference (with any options elected by the author(s) and/or publisher) is displayed in the reproduction.

Proper form for an incorporation by reference is as follows:

Copyright (c) <year> by <author's name or designee>. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, vX.Y or later (the latest version is presently available at <http://www.opencontent.org/openpub/>)

The reference must be immediately followed with any options elected by the author(s) and/or publisher of the document (see section VI). Commercial redistribution of Open Publication-licensed material is permitted. Any publication in standard (paper) book form shall require the citation of the original publisher and author. The publisher and author's names shall appear on all outer surfaces of the book. On all outer surfaces of the book the original publisher's name shall be as large as the title of the work and cited as possessive with respect to the title.

11.19.2 II. COPYRIGHT

The copyright to each Open Publication is owned by its author(s) or designee.

11.19.3 III. SCOPE OF LICENSE

The following license terms apply to all Open Publication works, unless otherwise explicitly stated in the document.

Mere aggregation of Open Publication works or a portion of an Open Publication work with other works or programs on the same media shall not cause this license to apply to those other works. The aggregate work shall contain a notice specifying the inclusion of the Open Publication material and appropriate copyright notice.

SEVERABILITY. If any part of this license is found to be unenforceable in any jurisdiction, the remaining portions of the license remain in force.

NO WARRANTY. Open Publication works are licensed and provided "as is" without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or a warranty of non-infringement.

11.19.4 IV. REQUIREMENTS ON MODIFIED WORKS

All modified versions of documents covered by this license, including translations, anthologies, compilations and partial documents, must meet the following requirements:

1. The modified version must be labeled as such.
2. The person making the modifications must be identified and the modifications dated.
3. Acknowledgement of the original author and publisher if applicable must be retained according to normal academic citation practices.
4. The location of the original unmodified document must be identified.
5. The original author's (or authors') name(s) may not be used to assert or imply endorsement of the resulting document without the original author's (or authors') permission.

11.19.5 V. GOOD-PRACTICE RECOMMENDATIONS

In addition to the requirements of this license, it is requested from and strongly recommended of redistributors that:

1. If you are distributing Open Publication works on hardcopy or CD-ROM, you provide email notification to the authors of your intent to redistribute at least thirty days before your manuscript or media freeze, to give the authors time to provide updated documents. This notification should describe modifications, if any, made to the document.
2. All substantive modifications (including deletions) be either clearly marked up in the document or else described in an attachment to the document.
3. Finally, while it is not mandatory under this license, it is considered good form to offer a free copy of any hardcopy and CD-ROM expression of an Open Publication-licensed work to its author(s).

11.19.6 VI. LICENSE OPTIONS

The author(s) and/or publisher of an Open Publication-licensed document may elect certain options by appending language to the reference to or copy of the license. These options are considered part of the license instance and must be included with the license (or its incorporation by reference) in derived works.

A. To prohibit distribution of substantively modified versions without the explicit permission of the author(s). "Substantive modification" is defined as a change to the semantic content of the document, and excludes mere changes in format or typographical corrections.

To accomplish this, add the phrase 'Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder.' to the license reference or copy.

B. To prohibit any publication of this work or derivative works in whole or in part in standard (paper) book form for commercial purposes is prohibited unless prior permission is obtained from the

copyright holder.

To accomplish this, add the phrase 'Distribution of the work or derivative of the work in any standard (paper) book form is prohibited unless prior permission is obtained from the copyright holder.' to the license reference or copy.

12 Version HTML Help

13 Index

13.1 Index des fonctions

- A
- ♦ [maxdb->auto_commit](#)
- _
- ♦ [__halt_compiler](#)
- A
- ♦ [abs](#)
- ♦ [acos](#)
- ♦ [acosh](#)
- ♦ [addslashes](#)
- ♦ [addslashes](#)
- ♦ [aggregate](#)
- ♦ [aggregate_info](#)
- ♦ [aggregate_methods](#)
- ♦ [aggregate_methods_by_list](#)
- ♦ [aggregate_methods_by_regexp](#)
- ♦ [aggregate_properties](#)
- ♦ [aggregate_properties_by_list](#)
- ♦ [aggregate_properties_by_regexp](#)
- ♦ [aggregation_info](#)
- ♦ [apache_child_terminate](#)
- ♦ [apache_get_modules](#)
- ♦ [apache_get_version](#)
- ♦ [apache_getenv](#)
- ♦ [apache_lookup_uri](#)
- ♦ [apache_note](#)
- ♦ [apache_request_headers](#)
- ♦ [apache_reset_timeout](#)
- ♦ [apache_response_headers](#)
- ♦ [apache_setenv](#)
- ♦ [apd_breakpoint](#)
- ♦ [apd_callstack](#)
- ♦ [apd_clunk](#)
- ♦ [apd_continue](#)
- ♦ [apd_croak](#)
- ♦ [apd_dump_function_table](#)
- ♦ [apd_dump_persistent_resources](#)
- ♦ [apd_dump_regular_resources](#)
- ♦ [apd_echo](#)
- ♦ [apd_get_active_symbols](#)
- ♦ [apd_set_pprof_trace](#)
- ♦ [apd_set_session](#)
- ♦ [apd_set_session_trace](#)
- ♦ [apd_set_socket_session_trace](#)
- ♦ [array](#)
- ♦ [array_change_key_case](#)
- ♦ [array_chunk](#)
- ♦ [array_combine](#)
- ♦ [array_count_values](#)

- ◆ [array_diff](#)
- ◆ [array_diff_assoc](#)
- ◆ [array_diff_key](#)
- ◆ [array_diff_uassoc](#)
- ◆ [array_diff_ukey](#)
- ◆ [array_fill](#)
- ◆ [array_filter](#)
- ◆ [array_flip](#)
- ◆ [array_intersect](#)
- ◆ [array_intersect_assoc](#)
- ◆ [array_intersect_key](#)
- ◆ [array_intersect_uassoc](#)
- ◆ [array_intersect_ukey](#)
- ◆ [array_key_exists](#)
- ◆ [array_keys](#)
- ◆ [array_map](#)
- ◆ [array_merge](#)
- ◆ [array_merge_recursive](#)
- ◆ [array_multisort](#)
- ◆ [array_pad](#)
- ◆ [array_pop](#)
- ◆ [array_product](#)
- ◆ [array_push](#)
- ◆ [array_rand](#)
- ◆ [array_reduce](#)
- ◆ [array_reverse](#)
- ◆ [array_search](#)
- ◆ [array_shift](#)
- ◆ [array_slice](#)
- ◆ [array_splice](#)
- ◆ [array_sum](#)
- ◆ [array_udiff](#)
- ◆ [array_udiff_assoc](#)
- ◆ [array_udiff_uassoc](#)
- ◆ [array_uintersect](#)
- ◆ [array_uintersect_assoc](#)
- ◆ [array_uintersect_uassoc](#)
- ◆ [array_unique](#)
- ◆ [array_unshift](#)
- ◆ [array_values](#)
- ◆ [array_walk](#)
- ◆ [array_walk_recursive](#)
- ◆ [arrayiterator::current](#)
- ◆ [arrayiterator::key](#)
- ◆ [arrayiterator::next](#)
- ◆ [arrayiterator::rewind](#)
- ◆ [arrayiterator::seek](#)
- ◆ [arrayiterator::valid](#)
- ◆ [arrayobject::__construct](#)
- ◆ [arrayobject::append](#)
- ◆ [arrayobject::count](#)
- ◆ [arrayobject::getiterator](#)
- ◆ [arrayobject::offsetexists](#)
- ◆ [arrayobject::offsetget](#)

- ◆ [arrayobject::offsetset](#)
- ◆ [arrayobject::offsetunset](#)
- ◆ [arsort](#)
- ◆ [ascii2ebcdic](#)
- ◆ [asin](#)
- ◆ [asinh](#)
- ◆ [asort](#)
- ◆ [aspell_check](#)
- ◆ [aspell_check_raw](#)
- ◆ [aspell_new](#)
- ◆ [aspell_suggest](#)
- ◆ [assert](#)
- ◆ [assert_options](#)
- ◆ [atan](#)
- ◆ [atan2](#)
- ◆ [atanh](#)
- **B**
- - ◆ [base64_decode](#)
 - ◆ [base64_encode](#)
 - ◆ [base_convert](#)
 - ◆ [basename](#)
 - ◆ [bcadd](#)
 - ◆ [bccomp](#)
 - ◆ [bcdiv](#)
 - ◆ [bcmod](#)
 - ◆ [bcmul](#)
 - ◆ [bcompiler_load](#)
 - ◆ [bcompiler_load_exe](#)
 - ◆ [bcompiler_parse_class](#)
 - ◆ [bcompiler_read](#)
 - ◆ [bcompiler_write_class](#)
 - ◆ [bcompiler_write_constant](#)
 - ◆ [bcompiler_write_exe_footer](#)
 - ◆ [bcompiler_write_file](#)
 - ◆ [bcompiler_write_footer](#)
 - ◆ [bcompiler_write_function](#)
 - ◆ [bcompiler_write_functions_from_file](#)
 - ◆ [bcompiler_write_header](#)
 - ◆ [bcpow](#)
 - ◆ [bcpowmod](#)
 - ◆ [bcscale](#)
 - ◆ [bcsqrt](#)
 - ◆ [bcsub](#)
 - ◆ [bin2hex](#)
 - ◆ [bind_textdomain_codeset](#)
 - ◆ [bindec](#)
 - ◆ [bindtextdomain](#)
 - ◆ [bzclose](#)
 - ◆ [bzcompress](#)
 - ◆ [bzdecompress](#)
 - ◆ [bzerrno](#)
 - ◆ [bzerror](#)
 - ◆ [bzerrstr](#)
 - ◆ [bzflush](#)

- ◆ [bzopen](#)
- ◆ [bzread](#)
- ◆ [bzwrite](#)
- **C**
- - ◆ [cachingiterator::__toString](#)
 - ◆ [cachingiterator::hasnext](#)
 - ◆ [cachingiterator::next](#)
 - ◆ [cachingiterator::rewind](#)
 - ◆ [cachingiterator::valid](#)
 - ◆ [cachingrecursiveiterator::getchildren](#)
 - ◆ [cachingrecursiveiterator::haschildren](#)
 - ◆ [cal_days_in_month](#)
 - ◆ [cal_from_jd](#)
 - ◆ [cal_info](#)
 - ◆ [cal_to_jd](#)
 - ◆ [call_user_func](#)
 - ◆ [call_user_func_array](#)
 - ◆ [call_user_method](#)
 - ◆ [call_user_method_array](#)
 - ◆ [ccvs_add](#)
 - ◆ [ccvs_auth](#)
 - ◆ [ccvs_command](#)
 - ◆ [ccvs_count](#)
 - ◆ [ccvs_delete](#)
 - ◆ [ccvs_done](#)
 - ◆ [ccvs_init](#)
 - ◆ [ccvs_lookup](#)
 - ◆ [ccvs_new](#)
 - ◆ [ccvs_report](#)
 - ◆ [ccvs_return](#)
 - ◆ [ccvs_reverse](#)
 - ◆ [ccvs_sale](#)
 - ◆ [ccvs_status](#)
 - ◆ [ccvs_textvalue](#)
 - ◆ [ccvs_void](#)
 - ◆ [ceil](#)
 - ◆ [chdir](#)
 - ◆ [checkdate](#)
 - ◆ [checkdnsrr](#)
 - ◆ [chgrp](#)
 - ◆ [chmod](#)
 - ◆ [chop](#)
 - ◆ [chown](#)
 - ◆ [chr](#)
 - ◆ [chroot](#)
 - ◆ [chunk_split](#)
 - ◆ [class_exists](#)
 - ◆ [class_implements](#)
 - ◆ [class_parents](#)
 - ◆ [classkit_import](#)
 - ◆ [classkit_method_add](#)
 - ◆ [classkit_method_copy](#)
 - ◆ [classkit_method_redefine](#)
 - ◆ [classkit_method_remove](#)

- ◆ [classkit_method_rename](#)
- ◆ [clearstatcache](#)
- ◆ [closedir](#)
- ◆ [closelog](#)
- ◆ [com](#)
- ◆ [com_addrf](#)
- ◆ [com_create_guid](#)
- ◆ [com_event_sink](#)
- ◆ [com_get](#)
- ◆ [com_get_active_object](#)
- ◆ [com_invoke](#)
- ◆ [com_isenum](#)
- ◆ [com_load](#)
- ◆ [com_load_typelib](#)
- ◆ [com_message_pump](#)
- ◆ [com_print_typeinfo](#)
- ◆ [com_propget](#)
- ◆ [com_propput](#)
- ◆ [com_propset](#)
- ◆ [com_release](#)
- ◆ [com_set](#)
- ◆ [compact](#)
- ◆ [connection_aborted](#)
- ◆ [connection_status](#)
- ◆ [connection_timeout](#)
- ◆ [constant](#)
- ◆ [convert_cyr_string](#)
- ◆ [convert_uudecode](#)
- ◆ [convert_uuencode](#)
- ◆ [copy](#)
- ◆ [cos](#)
- ◆ [cosh](#)
- ◆ [count](#)
- ◆ [count_chars](#)
- ◆ [cpdf_add_annotation](#)
- ◆ [cpdf_add_outline](#)
- ◆ [cpdf_arc](#)
- ◆ [cpdf_begin_text](#)
- ◆ [cpdf_circle](#)
- ◆ [cpdf_clip](#)
- ◆ [cpdf_close](#)
- ◆ [cpdf_closepath](#)
- ◆ [cpdf_closepath_fill_stroke](#)
- ◆ [cpdf_closepath_stroke](#)
- ◆ [cpdf_continue_text](#)
- ◆ [cpdf_curveto](#)
- ◆ [cpdf_end_text](#)
- ◆ [cpdf_fill](#)
- ◆ [cpdf_fill_stroke](#)
- ◆ [cpdf_finalize](#)
- ◆ [cpdf_finalize_page](#)
- ◆ [cpdf_global_set_document_limits](#)
- ◆ [cpdf_import_jpeg](#)
- ◆ [cpdf_lineto](#)

- ◆ [cpdf_moveto](#)
- ◆ [cpdf_newpath](#)
- ◆ [cpdf_open](#)
- ◆ [cpdf_output_buffer](#)
- ◆ [cpdf_page_init](#)
- ◆ [cpdf_place_inline_image](#)
- ◆ [cpdf_rect](#)
- ◆ [cpdf_restore](#)
- ◆ [cpdf_rlineto](#)
- ◆ [cpdf_rmoveto](#)
- ◆ [cpdf_rotate](#)
- ◆ [cpdf_rotate_text](#)
- ◆ [cpdf_save](#)
- ◆ [cpdf_save_to_file](#)
- ◆ [cpdf_scale](#)
- ◆ [cpdf_set_action_url](#)
- ◆ [cpdf_set_char_spacing](#)
- ◆ [cpdf_set_creator](#)
- ◆ [cpdf_set_current_page](#)
- ◆ [cpdf_set_font](#)
- ◆ [cpdf_set_font_directories](#)
- ◆ [cpdf_set_font_map_file](#)
- ◆ [cpdf_set_horiz_scaling](#)
- ◆ [cpdf_set_keywords](#)
- ◆ [cpdf_set_leading](#)
- ◆ [cpdf_set_page_animation](#)
- ◆ [cpdf_set_subject](#)
- ◆ [cpdf_set_text_matrix](#)
- ◆ [cpdf_set_text_pos](#)
- ◆ [cpdf_set_text_rendering](#)
- ◆ [cpdf_set_text_rise](#)
- ◆ [cpdf_set_title](#)
- ◆ [cpdf_set_viewer_preferences](#)
- ◆ [cpdf_set_word_spacing](#)
- ◆ [cpdf_setdash](#)
- ◆ [cpdf_setflat](#)
- ◆ [cpdf_setgray](#)
- ◆ [cpdf_setgray_fill](#)
- ◆ [cpdf_setgray_stroke](#)
- ◆ [cpdf_setlinecap](#)
- ◆ [cpdf_setlinejoin](#)
- ◆ [cpdf_setlinewidth](#)
- ◆ [cpdf_setmiterlimit](#)
- ◆ [cpdf_setrgbcolor](#)
- ◆ [cpdf_setrgbcolor_fill](#)
- ◆ [cpdf_setrgbcolor_stroke](#)
- ◆ [cpdf_show](#)
- ◆ [cpdf_show_xy](#)
- ◆ [cpdf_stringwidth](#)
- ◆ [cpdf_stroke](#)
- ◆ [cpdf_text](#)
- ◆ [cpdf_translate](#)
- ◆ [crack_check](#)
- ◆ [crack_closedict](#)

- ◆ [crack_getlastmessage](#)
- ◆ [crack_opendict](#)
- ◆ [crc32](#)
- ◆ [create_function](#)
- ◆ [crypt](#)
- ◆ [ctype_alnum](#)
- ◆ [ctype_alpha](#)
- ◆ [ctype_cntrl](#)
- ◆ [ctype_digit](#)
- ◆ [ctype_graph](#)
- ◆ [ctype_lower](#)
- ◆ [ctype_print](#)
- ◆ [ctype_punct](#)
- ◆ [ctype_space](#)
- ◆ [ctype_upper](#)
- ◆ [ctype_xdigit](#)
- ◆ [curl_close](#)
- ◆ [curl_copy_handle](#)
- ◆ [curl_errno](#)
- ◆ [curl_error](#)
- ◆ [curl_exec](#)
- ◆ [curl_getinfo](#)
- ◆ [curl_init](#)
- ◆ [curl_multi_add_handle](#)
- ◆ [curl_multi_close](#)
- ◆ [curl_multi_exec](#)
- ◆ [curl_multi_getcontent](#)
- ◆ [curl_multi_info_read](#)
- ◆ [curl_multi_init](#)
- ◆ [curl_multi_remove_handle](#)
- ◆ [curl_multi_select](#)
- ◆ [curl_setopt](#)
- ◆ [curl_version](#)
- ◆ [current](#)
- ◆ [cybercash_base64_decode](#)
- ◆ [cybercash_base64_encode](#)
- ◆ [cybercash_decr](#)
- ◆ [cybercash_encr](#)
- ◆ [cybermut_creeformulairsecm](#)
- ◆ [cybermut_creerreponsecm](#)
- ◆ [cybermut_testmac](#)
- ◆ [cyrus_authenticate](#)
- ◆ [cyrus_bind](#)
- ◆ [cyrus_close](#)
- ◆ [cyrus_connect](#)
- ◆ [cyrus_query](#)
- ◆ [cyrus_unbind](#)
- **D**
- - ◆ [date](#)
 - ◆ [date_default_timezone_get](#)
 - ◆ [date_default_timezone_set](#)
 - ◆ [date_sunrise](#)
 - ◆ [date_sunset](#)
 - ◆ [dba_close](#)

- ◆ [dba_delete](#)
- ◆ [dba_exists](#)
- ◆ [dba_fetch](#)
- ◆ [dba_firstkey](#)
- ◆ [dba_handlers](#)
- ◆ [dba_insert](#)
- ◆ [dba_key_split](#)
- ◆ [dba_list](#)
- ◆ [dba_nextkey](#)
- ◆ [dba_open](#)
- ◆ [dba_optimize](#)
- ◆ [dba_popen](#)
- ◆ [dba_replace](#)
- ◆ [dba_sync](#)
- ◆ [dbase_add_record](#)
- ◆ [dbase_close](#)
- ◆ [dbase_create](#)
- ◆ [dbase_delete_record](#)
- ◆ [dbase_get_header_info](#)
- ◆ [dbase_get_record](#)
- ◆ [dbase_get_record_with_names](#)
- ◆ [dbase_numfields](#)
- ◆ [dbase_numrecords](#)
- ◆ [dbase_open](#)
- ◆ [dbase_pack](#)
- ◆ [dbase_replace_record](#)
- ◆ [dblist](#)
- ◆ [dbmclose](#)
- ◆ [dbmdelete](#)
- ◆ [dbmexists](#)
- ◆ [dbmfetch](#)
- ◆ [dbmfirstkey](#)
- ◆ [dbminsert](#)
- ◆ [dbmnextkey](#)
- ◆ [dbmopen](#)
- ◆ [dbmreplace](#)
- ◆ [dbplus_add](#)
- ◆ [dbplus_aql](#)
- ◆ [dbplus_chdir](#)
- ◆ [dbplus_close](#)
- ◆ [dbplus_curr](#)
- ◆ [dbplus_errcode](#)
- ◆ [dbplus_errno](#)
- ◆ [dbplus_find](#)
- ◆ [dbplus_first](#)
- ◆ [dbplus_flush](#)
- ◆ [dbplus_freealllocks](#)
- ◆ [dbplus_freelock](#)
- ◆ [dbplus_freerlocks](#)
- ◆ [dbplus_getlock](#)
- ◆ [dbplus_getunique](#)
- ◆ [dbplus_info](#)
- ◆ [dbplus_last](#)
- ◆ [dbplus_lockrel](#)

- ◆ [dbplus_next](#)
- ◆ [dbplus_open](#)
- ◆ [dbplus_prev](#)
- ◆ [dbplus_rchperm](#)
- ◆ [dbplus_rcreate](#)
- ◆ [dbplus_rrtexact](#)
- ◆ [dbplus_rrtlike](#)
- ◆ [dbplus_resolve](#)
- ◆ [dbplus_restorepos](#)
- ◆ [dbplus_rkeys](#)
- ◆ [dbplus_ropen](#)
- ◆ [dbplus_rquery](#)
- ◆ [dbplus_rrename](#)
- ◆ [dbplus_rsecindex](#)
- ◆ [dbplus_runlink](#)
- ◆ [dbplus_rzap](#)
- ◆ [dbplus_savepos](#)
- ◆ [dbplus_setindex](#)
- ◆ [dbplus_setindexbynumber](#)
- ◆ [dbplus_sql](#)
- ◆ [dbplus_tcl](#)
- ◆ [dbplus_tremove](#)
- ◆ [dbplus_undo](#)
- ◆ [dbplus_undoprepere](#)
- ◆ [dbplus_unlockrel](#)
- ◆ [dbplus_unselect](#)
- ◆ [dbplus_update](#)
- ◆ [dbplus_xlockrel](#)
- ◆ [dbplus_xunlockrel](#)
- ◆ [dbx_close](#)
- ◆ [dbx_compare](#)
- ◆ [dbx_connect](#)
- ◆ [dbx_error](#)
- ◆ [dbx_escape_string](#)
- ◆ [dbx_fetch_row](#)
- ◆ [dbx_query](#)
- ◆ [dbx_sort](#)
- ◆ [dcgettext](#)
- ◆ [dcngettext](#)
- ◆ [deaggregate](#)
- ◆ [debug_backtrace](#)
- ◆ [debug_print_backtrace](#)
- ◆ [debug_zval_dump](#)
- ◆ [debugger_off](#)
- ◆ [debugger_on](#)
- ◆ [decbin](#)
- ◆ [dechex](#)
- ◆ [decoct](#)
- ◆ [define](#)
- ◆ [define_syslog_variables](#)
- ◆ [defined](#)
- ◆ [deg2rad](#)
- ◆ [delete](#)
- ◆ [dgettext](#)

- ◆ [die](#)
- ◆ [dio_close](#)
- ◆ [dio_fcntl](#)
- ◆ [dio_open](#)
- ◆ [dio_read](#)
- ◆ [dio_seek](#)
- ◆ [dio_stat](#)
- ◆ [dio_tcsetattr](#)
- ◆ [dio_truncate](#)
- ◆ [dio_write](#)
- ◆ [dir](#)
- ◆ [directoryiterator::__construct](#)
- ◆ [directoryiterator::current](#)
- ◆ [directoryiterator::getatime](#)
- ◆ [directoryiterator::getchildren](#)
- ◆ [directoryiterator::getctime](#)
- ◆ [directoryiterator::getfilename](#)
- ◆ [directoryiterator::getgroup](#)
- ◆ [directoryiterator::getinode](#)
- ◆ [directoryiterator::getmtime](#)
- ◆ [directoryiterator::getowner](#)
- ◆ [directoryiterator::getpath](#)
- ◆ [directoryiterator::getpathname](#)
- ◆ [directoryiterator::getperms](#)
- ◆ [directoryiterator::getsize](#)
- ◆ [directoryiterator::gettype](#)
- ◆ [directoryiterator::isdir](#)
- ◆ [directoryiterator::isdot](#)
- ◆ [directoryiterator::isexecutable](#)
- ◆ [directoryiterator::isfile](#)
- ◆ [directoryiterator::islink](#)
- ◆ [directoryiterator::isreadable](#)
- ◆ [directoryiterator::iswritable](#)
- ◆ [directoryiterator::key](#)
- ◆ [directoryiterator::next](#)
- ◆ [directoryiterator::rewind](#)
- ◆ [directoryiterator::valid](#)
- ◆ [dirname](#)
- ◆ [disk_free_space](#)
- ◆ [disk_total_space](#)
- ◆ [diskfreespace](#)
- ◆ [dl](#)
- ◆ [dgettext](#)
- ◆ [dns_check_record](#)
- ◆ [dns_get_mx](#)
- ◆ [dns_get_record](#)
- ◆ [dom_import_simplexml](#)
- ◆ [domattr->__construct\(\)](#)
- ◆ [domattr->isid\(\)](#)
- ◆ [domattribute->name](#)
- ◆ [domattribute->set_value](#)
- ◆ [domattribute->specified](#)
- ◆ [domattribute->value](#)
- ◆ [domcharacterdata->appenddata\(\)](#)

- ◆ [domcharacterdata->deletedata\(\)](#)
- ◆ [domcharacterdata->insertdata\(\)](#)
- ◆ [domcharacterdata->replacedata\(\)](#)
- ◆ [domcharacterdata->substringdata\(\)](#)
- ◆ [domcomment->__construct\(\)](#)
- ◆ [domdocument->__construct\(\)](#)
- ◆ [domdocument->add_root](#)
- ◆ [domdocument->create_attribute](#)
- ◆ [domdocument->create_cdata_section](#)
- ◆ [domdocument->create_comment](#)
- ◆ [domdocument->create_element](#)
- ◆ [domdocument->create_element_ns](#)
- ◆ [domdocument->create_entity_reference](#)
- ◆ [domdocument->create_processing_instruction](#)
- ◆ [domdocument->create_text_node](#)
- ◆ [domdocument->createattribute\(\)](#)
- ◆ [domdocument->createattributens\(\)](#)
- ◆ [domdocument->createcdatasection\(\)](#)
- ◆ [domdocument->createcomment\(\)](#)
- ◆ [domdocument->createdocumentfragment\(\)](#)
- ◆ [domdocument->createelement\(\)](#)
- ◆ [domdocument->createelementns\(\)](#)
- ◆ [domdocument->createentityreference\(\)](#)
- ◆ [domdocument->createprocessinginstruction\(\)](#)
- ◆ [domdocument->createtextnode\(\)](#)
- ◆ [domdocument->doctype](#)
- ◆ [domdocument->document_element](#)
- ◆ [domdocument->dump_file](#)
- ◆ [domdocument->dump_mem](#)
- ◆ [domdocument->get_element_by_id](#)
- ◆ [domdocument->get_elements_by_tagname](#)
- ◆ [domdocument->getelementbyid\(\)](#)
- ◆ [domdocument->getelementsbytagname\(\)](#)
- ◆ [domdocument->getelementsbytagname_ns\(\)](#)
- ◆ [domdocument->html_dump_mem](#)
- ◆ [domdocument->importnode\(\)](#)
- ◆ [domdocument->load\(\)](#)
- ◆ [domdocument->loadhtml\(\)](#)
- ◆ [domdocument->loadhtmlfile\(\)](#)
- ◆ [domdocument->loadxml\(\)](#)
- ◆ [domdocument->normalize\(\)](#)
- ◆ [domdocument->relaxngvalidate\(\)](#)
- ◆ [domdocument->relaxngvalidatesource\(\)](#)
- ◆ [domdocument->save\(\)](#)
- ◆ [domdocument->savehtml\(\)](#)
- ◆ [domdocument->savehtmlfile\(\)](#)
- ◆ [domdocument->savexml\(\)](#)
- ◆ [domdocument->schemavalidate\(\)](#)
- ◆ [domdocument->schemavalidatesource\(\)](#)
- ◆ [domdocument->validate\(\)](#)
- ◆ [domdocument->xinclude](#)
- ◆ [domdocument->xinclude\(\)](#)
- ◆ [domdocumenttype->entities\(\)](#)
- ◆ [domdocumenttype->internal_subset\(\)](#)

- ◆ [domdocumenttype->name\(\)](#)
- ◆ [domdocumenttype->notations\(\)](#)
- ◆ [domdocumenttype->public_id\(\)](#)
- ◆ [domdocumenttype->system_id\(\)](#)
- ◆ [domelement->__construct\(\)](#)
- ◆ [domelement->get_attribute\(\)](#)
- ◆ [domelement->get_attribute_node\(\)](#)
- ◆ [domelement->get_elements_by_tagname\(\)](#)
- ◆ [domelement->getattribute\(\)](#)
- ◆ [domelement->getattributenode\(\)](#)
- ◆ [domelement->getattributenodens\(\)](#)
- ◆ [domelement->getattributens\(\)](#)
- ◆ [domelement->getelementsbytagname\(\)](#)
- ◆ [domelement->getelementsbytagnamens\(\)](#)
- ◆ [domelement->has_attribute\(\)](#)
- ◆ [domelement->hasattribute\(\)](#)
- ◆ [domelement->hasattributens\(\)](#)
- ◆ [domelement->remove_attribute\(\)](#)
- ◆ [domelement->removeattribute\(\)](#)
- ◆ [domelement->removeattributenode\(\)](#)
- ◆ [domelement->removeattributens\(\)](#)
- ◆ [domelement->set_attribute\(\)](#)
- ◆ [domelement->setattribute\(\)](#)
- ◆ [domelement->setattributenode\(\)](#)
- ◆ [domelement->setattributenodens\(\)](#)
- ◆ [domelement->setattributens\(\)](#)
- ◆ [domelement->tagname\(\)](#)
- ◆ [domentityreference->__construct\(\)](#)
- ◆ [domimplementation->__construct\(\)](#)
- ◆ [domimplementation->createdocument\(\)](#)
- ◆ [domimplementation->createdocumenttype\(\)](#)
- ◆ [domimplementation->hasfeature\(\)](#)
- ◆ [domnamednodemap->getnameditem\(\)](#)
- ◆ [domnamednodemap->getnameditemns\(\)](#)
- ◆ [domnamednodemap->item\(\)](#)
- ◆ [domnode->add_namespace](#)
- ◆ [domnode->append_child](#)
- ◆ [domnode->append_sibling](#)
- ◆ [domnode->appendchild\(\)](#)
- ◆ [domnode->attributes](#)
- ◆ [domnode->child_nodes](#)
- ◆ [domnode->clone_node](#)
- ◆ [domnode->clonenode\(\)](#)
- ◆ [domnode->dump_node](#)
- ◆ [domnode->first_child](#)
- ◆ [domnode->get_content](#)
- ◆ [domnode->has_attributes](#)
- ◆ [domnode->has_child_nodes](#)
- ◆ [domnode->hasattributes\(\)](#)
- ◆ [domnode->haschildnodes\(\)](#)
- ◆ [domnode->insert_before](#)
- ◆ [domnode->insertbefore\(\)](#)
- ◆ [domnode->is_blank_node](#)
- ◆ [domnode->issamenode\(\)](#)

- ◆ [domnode->issupported\(\)](#)
- ◆ [domnode->last_child](#)
- ◆ [domnode->lookupnamespaceuri\(\)](#)
- ◆ [domnode->lookupprefix\(\)](#)
- ◆ [domnode->next_sibling](#)
- ◆ [domnode->node_name](#)
- ◆ [domnode->node_type](#)
- ◆ [domnode->node_value](#)
- ◆ [domnode->normalize\(\)](#)
- ◆ [domnode->owner_document](#)
- ◆ [domnode->parent_node](#)
- ◆ [domnode->prefix](#)
- ◆ [domnode->previous_sibling](#)
- ◆ [domnode->remove_child](#)
- ◆ [domnode->removechild\(\)](#)
- ◆ [domnode->replace_child](#)
- ◆ [domnode->replace_node](#)
- ◆ [domnode->replacechild\(\)](#)
- ◆ [domnode->set_content](#)
- ◆ [domnode->set_name](#)
- ◆ [domnode->set_namespace](#)
- ◆ [domnode->unlink_node](#)
- ◆ [domodelist->item\(\)](#)
- ◆ [domprocessinginstruction->__construct\(\)](#)
- ◆ [domprocessinginstruction->data](#)
- ◆ [domprocessinginstruction->target](#)
- ◆ [domtext->__construct\(\)](#)
- ◆ [domtext->iswhitespaceinelementcontent\(\)](#)
- ◆ [domtext->splittext\(\)](#)
- ◆ [domxml_new_doc](#)
- ◆ [domxml_open_file](#)
- ◆ [domxml_open_mem](#)
- ◆ [domxml_version](#)
- ◆ [domxml_xmltree](#)
- ◆ [domxml_xslt_stylesheet](#)
- ◆ [domxml_xslt_stylesheet_doc](#)
- ◆ [domxml_xslt_stylesheet_file](#)
- ◆ [domxml_xslt_version](#)
- ◆ [domxpath->__construct\(\)](#)
- ◆ [domxpath->evaluate\(\)](#)
- ◆ [domxpath->query\(\)](#)
- ◆ [domxpath->registernamespace\(\)](#)
- ◆ [domxsltstylesheet->process\(\)](#)
- ◆ [domxsltstylesheet->result_dump_file\(\)](#)
- ◆ [domxsltstylesheet->result_dump_mem\(\)](#)
- ◆ [dotnet](#)
- ◆ [dotnet_load](#)
- ◆ [doubleval](#)
- **E**
- ◆ [each](#)
- ◆ [easter_date](#)
- ◆ [easter_days](#)
- ◆ [ebcdic2ascii](#)
- ◆ [echo](#)

- ◆ [empty](#)
- ◆ [end](#)
- ◆ [ereg](#)
- ◆ [ereg_replace](#)
- ◆ [eregi](#)
- ◆ [eregi_replace](#)
- ◆ [error_log](#)
- ◆ [error_reporting](#)
- ◆ [escapeshellarg](#)
- ◆ [escapeshellcmd](#)
- ◆ [eval](#)
- ◆ [exec](#)
- ◆ [exif_imagetype](#)
- ◆ [exif_read_data](#)
- ◆ [exif_tagname](#)
- ◆ [exif_thumbnail](#)
- ◆ [exit](#)
- ◆ [exp](#)
- ◆ [explode](#)
- ◆ [expm1](#)
- ◆ [extension_loaded](#)
- ◆ [extract](#)
- ◆ [ezmlm_hash](#)
- **F**
- - ◆ [fam_cancel_monitor](#)
 - ◆ [fam_close](#)
 - ◆ [fam_monitor_collection](#)
 - ◆ [fam_monitor_directory](#)
 - ◆ [fam_monitor_file](#)
 - ◆ [fam_next_event](#)
 - ◆ [fam_open](#)
 - ◆ [fam_pending](#)
 - ◆ [fam_resume_monitor](#)
 - ◆ [fam_suspend_monitor](#)
 - ◆ [fbsql_affected_rows](#)
 - ◆ [fbsql_autocommit](#)
 - ◆ [fbsql_blob_size](#)
 - ◆ [fbsql_change_user](#)
 - ◆ [fbsql_clob_size](#)
 - ◆ [fbsql_close](#)
 - ◆ [fbsql_commit](#)
 - ◆ [fbsql_connect](#)
 - ◆ [fbsql_create_blob](#)
 - ◆ [fbsql_create_clob](#)
 - ◆ [fbsql_create_db](#)
 - ◆ [fbsql_data_seek](#)
 - ◆ [fbsql_database](#)
 - ◆ [fbsql_database_password](#)
 - ◆ [fbsql_db_query](#)
 - ◆ [fbsql_db_status](#)
 - ◆ [fbsql_drop_db](#)
 - ◆ [fbsql_erro](#)
 - ◆ [fbsql_error](#)
 - ◆ [fbsql_fetch_array](#)

- ◆ [fbsql_fetch_assoc](#)
- ◆ [fbsql_fetch_field](#)
- ◆ [fbsql_fetch_lengths](#)
- ◆ [fbsql_fetch_object](#)
- ◆ [fbsql_fetch_row](#)
- ◆ [fbsql_field_flags](#)
- ◆ [fbsql_field_len](#)
- ◆ [fbsql_field_name](#)
- ◆ [fbsql_field_seek](#)
- ◆ [fbsql_field_table](#)
- ◆ [fbsql_field_type](#)
- ◆ [fbsql_free_result](#)
- ◆ [fbsql_get_autostart_info](#)
- ◆ [fbsql_hostname](#)
- ◆ [fbsql_insert_id](#)
- ◆ [fbsql_list_dbs](#)
- ◆ [fbsql_list_fields](#)
- ◆ [fbsql_list_tables](#)
- ◆ [fbsql_next_result](#)
- ◆ [fbsql_num_fields](#)
- ◆ [fbsql_num_rows](#)
- ◆ [fbsql_password](#)
- ◆ [fbsql_pconnect](#)
- ◆ [fbsql_query](#)
- ◆ [fbsql_read_blob](#)
- ◆ [fbsql_read_clob](#)
- ◆ [fbsql_result](#)
- ◆ [fbsql_rollback](#)
- ◆ [fbsql_select_db](#)
- ◆ [fbsql_set_lob_mode](#)
- ◆ [fbsql_set_password](#)
- ◆ [fbsql_set_transaction](#)
- ◆ [fbsql_start_db](#)
- ◆ [fbsql_stop_db](#)
- ◆ [fbsql_tablename](#)
- ◆ [fbsql_username](#)
- ◆ [fbsql_warnings](#)
- ◆ [fclose](#)
- ◆ [fdf_add_doc_javascript](#)
- ◆ [fdf_add_template](#)
- ◆ [fdf_close](#)
- ◆ [fdf_create](#)
- ◆ [fdf_enum_values](#)
- ◆ [fdf_errno](#)
- ◆ [fdf_error](#)
- ◆ [fdf_get_ap](#)
- ◆ [fdf_get_attachment](#)
- ◆ [fdf_get_encoding](#)
- ◆ [fdf_get_file](#)
- ◆ [fdf_get_flags](#)
- ◆ [fdf_get_opt](#)
- ◆ [fdf_get_status](#)
- ◆ [fdf_get_value](#)
- ◆ [fdf_get_version](#)

- ◆ [fdf_header](#)
- ◆ [fdf_next_field_name](#)
- ◆ [fdf_open](#)
- ◆ [fdf_open_string](#)
- ◆ [fdf_remove_item](#)
- ◆ [fdf_save](#)
- ◆ [fdf_save_string](#)
- ◆ [fdf_set_ap](#)
- ◆ [fdf_set_encoding](#)
- ◆ [fdf_set_file](#)
- ◆ [fdf_set_flags](#)
- ◆ [fdf_set_javascript_action](#)
- ◆ [fdf_set_on_import_javascript](#)
- ◆ [fdf_set_opt](#)
- ◆ [fdf_set_status](#)
- ◆ [fdf_set_submit_form_action](#)
- ◆ [fdf_set_target_frame](#)
- ◆ [fdf_set_value](#)
- ◆ [fdf_set_version](#)
- ◆ [feof](#)
- ◆ [fflush](#)
- ◆ [fgetc](#)
- ◆ [fgetcsv](#)
- ◆ [fgets](#)
- ◆ [fgetss](#)
- ◆ [file](#)
- ◆ [file_exists](#)
- ◆ [file_get_contents](#)
- ◆ [file_put_contents](#)
- ◆ [fileatime](#)
- ◆ [filectime](#)
- ◆ [filegroup](#)
- ◆ [fileinode](#)
- ◆ [filemtime](#)
- ◆ [fileowner](#)
- ◆ [fileperms](#)
- ◆ [filepro](#)
- ◆ [filepro_fieldcount](#)
- ◆ [filepro_fieldname](#)
- ◆ [filepro_fieldtype](#)
- ◆ [filepro_fieldwidth](#)
- ◆ [filepro_retrieve](#)
- ◆ [filepro_rowcount](#)
- ◆ [filesize](#)
- ◆ [filetype](#)
- ◆ [filteriterator::current](#)
- ◆ [filteriterator::getinneriterator](#)
- ◆ [filteriterator::key](#)
- ◆ [filteriterator::next](#)
- ◆ [filteriterator::rewind](#)
- ◆ [filteriterator::valid](#)
- ◆ [floatval](#)
- ◆ [flock](#)
- ◆ [floor](#)

- ◆ [flush](#)
- ◆ [fmod](#)
- ◆ [fnmatch](#)
- ◆ [fopen](#)
- ◆ [fpassthru](#)
- ◆ [fprintf](#)
- ◆ [fputcsv](#)
- ◆ [fputs](#)
- ◆ [fread](#)
- ◆ [frenchtojd](#)
- ◆ [fribidi_log2vis](#)
- ◆ [fscanf](#)
- ◆ [fseek](#)
- ◆ [fsockopen](#)
- ◆ [fstat](#)
- ◆ [ftell](#)
- ◆ [ftok](#)
- ◆ [ftp_alloc](#)
- ◆ [ftp_cdup](#)
- ◆ [ftp_chdir](#)
- ◆ [ftp_chmod](#)
- ◆ [ftp_close](#)
- ◆ [ftp_connect](#)
- ◆ [ftp_delete](#)
- ◆ [ftp_exec](#)
- ◆ [ftp_fget](#)
- ◆ [ftp_fput](#)
- ◆ [ftp_get](#)
- ◆ [ftp_get_option](#)
- ◆ [ftp_login](#)
- ◆ [ftp_mdtm](#)
- ◆ [ftp_mkdir](#)
- ◆ [ftp_nb_continue](#)
- ◆ [ftp_nb_fget](#)
- ◆ [ftp_nb_fput](#)
- ◆ [ftp_nb_get](#)
- ◆ [ftp_nb_put](#)
- ◆ [ftp_nlist](#)
- ◆ [ftp_pasv](#)
- ◆ [ftp_put](#)
- ◆ [ftp_pwd](#)
- ◆ [ftp_quit](#)
- ◆ [ftp_raw](#)
- ◆ [ftp_rawlist](#)
- ◆ [ftp_rename](#)
- ◆ [ftp_rmdir](#)
- ◆ [ftp_set_option](#)
- ◆ [ftp_site](#)
- ◆ [ftp_size](#)
- ◆ [ftp_ssl_connect](#)
- ◆ [ftp_systype](#)
- ◆ [truncate](#)
- ◆ [func_get_arg](#)
- ◆ [func_get_args](#)

- ◆ [func_num_args](#)
- ◆ [function_exists](#)
- ◆ [fwrite](#)
- **G**
- - ◆ [gd_info](#)
 - ◆ [get_browser](#)
 - ◆ [get_cfg_var](#)
 - ◆ [get_class](#)
 - ◆ [get_class_methods](#)
 - ◆ [get_class_vars](#)
 - ◆ [get_current_user](#)
 - ◆ [get_declared_classes](#)
 - ◆ [get_declared_interfaces](#)
 - ◆ [get_defined_constants](#)
 - ◆ [get_defined_functions](#)
 - ◆ [get_defined_vars](#)
 - ◆ [get_extension_funcs](#)
 - ◆ [get_headers](#)
 - ◆ [get_html_translation_table](#)
 - ◆ [get_include_path](#)
 - ◆ [get_included_files](#)
 - ◆ [get_loaded_extensions](#)
 - ◆ [get_magic_quotes_gpc](#)
 - ◆ [get_magic_quotes_runtime](#)
 - ◆ [get_meta_tags](#)
 - ◆ [get_object_vars](#)
 - ◆ [get_parent_class](#)
 - ◆ [get_required_files](#)
 - ◆ [get_resource_type](#)
 - ◆ [getallheaders](#)
 - ◆ [getcwd](#)
 - ◆ [getdate](#)
 - ◆ [getenv](#)
 - ◆ [gethostbyaddr](#)
 - ◆ [gethostbyname](#)
 - ◆ [gethostbyname_l](#)
 - ◆ [getimagesize](#)
 - ◆ [getlastmod](#)
 - ◆ [getmxrr](#)
 - ◆ [getmygid](#)
 - ◆ [getmyinode](#)
 - ◆ [getmypid](#)
 - ◆ [getmyuid](#)
 - ◆ [getopt](#)
 - ◆ [getprotobyname](#)
 - ◆ [getprotobynumber](#)
 - ◆ [getrandmax](#)
 - ◆ [getrusage](#)
 - ◆ [getservbyname](#)
 - ◆ [getservbyport](#)
 - ◆ [gettext](#)
 - ◆ [gettimeofday](#)
 - ◆ [gettype](#)
 - ◆ [glob](#)

- ◆ [gmdate](#)
- ◆ [gmmktime](#)
- ◆ [gmp_abs](#)
- ◆ [gmp_add](#)
- ◆ [gmp_and](#)
- ◆ [gmp_clrbit](#)
- ◆ [gmp_cmp](#)
- ◆ [gmp_com](#)
- ◆ [gmp_div](#)
- ◆ [gmp_div_q](#)
- ◆ [gmp_div_qr](#)
- ◆ [gmp_div_r](#)
- ◆ [gmp_divexact](#)
- ◆ [gmp_fact](#)
- ◆ [gmp_gcd](#)
- ◆ [gmp_gcdext](#)
- ◆ [gmp_hamdist](#)
- ◆ [gmp_init](#)
- ◆ [gmp_intval](#)
- ◆ [gmp_invert](#)
- ◆ [gmp_jacobi](#)
- ◆ [gmp_legendre](#)
- ◆ [gmp_mod](#)
- ◆ [gmp_mul](#)
- ◆ [gmp_neg](#)
- ◆ [gmp_or](#)
- ◆ [gmp_perfect_square](#)
- ◆ [gmp_popcount](#)
- ◆ [gmp_pow](#)
- ◆ [gmp_powm](#)
- ◆ [gmp_prob_prime](#)
- ◆ [gmp_random](#)
- ◆ [gmp_scan0](#)
- ◆ [gmp_scan1](#)
- ◆ [gmp_setbit](#)
- ◆ [gmp_sign](#)
- ◆ [gmp_sqrt](#)
- ◆ [gmp_sqrtrem](#)
- ◆ [gmp_strval](#)
- ◆ [gmp_sub](#)
- ◆ [gmp_xor](#)
- ◆ [gmstrftime](#)
- ◆ [gregoriantojd](#)
- ◆ [gzclose](#)
- ◆ [gzcompress](#)
- ◆ [gzdeflate](#)
- ◆ [gzencode](#)
- ◆ [gzeof](#)
- ◆ [gzfile](#)
- ◆ [gzgetc](#)
- ◆ [gzgets](#)
- ◆ [gzgetss](#)
- ◆ [gzinflate](#)
- ◆ [gzopen](#)

- ◆ [gzpassthru](#)
- ◆ [gzputs](#)
- ◆ [gzread](#)
- ◆ [gzrewind](#)
- ◆ [gzseek](#)
- ◆ [gztell](#)
- ◆ [gzuncompress](#)
- ◆ [gzwrite](#)
- H
 - ◆ [header](#)
 - ◆ [headers_list](#)
 - ◆ [headers_sent](#)
 - ◆ [hebreve](#)
 - ◆ [hebrevc](#)
 - ◆ [hexdec](#)
 - ◆ [highlight_file](#)
 - ◆ [highlight_string](#)
 - ◆ [html_entity_decode](#)
 - ◆ [htmlentities](#)
 - ◆ [htmlspecialchars](#)
 - ◆ [htmlspecialchars_decode](#)
 - ◆ [http_build_query](#)
 - ◆ [hw_api->checkin](#)
 - ◆ [hw_api->checkout](#)
 - ◆ [hw_api->children](#)
 - ◆ [hw_api->content](#)
 - ◆ [hw_api->copy](#)
 - ◆ [hw_api->dbstat](#)
 - ◆ [hw_api->dcstat](#)
 - ◆ [hw_api->dstanchors](#)
 - ◆ [hw_api->dstofsrcanchor](#)
 - ◆ [hw_api->find](#)
 - ◆ [hw_api->ftstat](#)
 - ◆ [hw_api->hwstat](#)
 - ◆ [hw_api->identify](#)
 - ◆ [hw_api->info](#)
 - ◆ [hw_api->insert](#)
 - ◆ [hw_api->insertanchor](#)
 - ◆ [hw_api->insertcollection](#)
 - ◆ [hw_api->insertdocument](#)
 - ◆ [hw_api->link](#)
 - ◆ [hw_api->lock](#)
 - ◆ [hw_api->move](#)
 - ◆ [hw_api->object](#)
 - ◆ [hw_api->objectbyanchor](#)
 - ◆ [hw_api->parents](#)
 - ◆ [hw_api->remove](#)
 - ◆ [hw_api->replace](#)
 - ◆ [hw_api->setcommittedversion](#)
 - ◆ [hw_api->srcanchors](#)
 - ◆ [hw_api->srcsofdst](#)
 - ◆ [hw_api->unlock](#)
 - ◆ [hw_api->user](#)
 - ◆ [hw_api->userlist](#)

- ◆ [hw_api_attribute](#)
- ◆ [hw_api_attribute->key](#)
- ◆ [hw_api_attribute->langdepvalue](#)
- ◆ [hw_api_attribute->value](#)
- ◆ [hw_api_attribute->values](#)
- ◆ [hw_api_content](#)
- ◆ [hw_api_content->mimetype](#)
- ◆ [hw_api_content->read](#)
- ◆ [hw_api_error->count](#)
- ◆ [hw_api_error->reason](#)
- ◆ [hw_api_object](#)
- ◆ [hw_api_object->assign](#)
- ◆ [hw_api_object->attreditable](#)
- ◆ [hw_api_object->count](#)
- ◆ [hw_api_object->insert](#)
- ◆ [hw_api_object->remove](#)
- ◆ [hw_api_object->title](#)
- ◆ [hw_api_object->value](#)
- ◆ [hw_api_reason->description](#)
- ◆ [hw_api_reason->type](#)
- ◆ [hw_array2objrec](#)
- ◆ [hw_changeobject](#)
- ◆ [hw_children](#)
- ◆ [hw_childrenobj](#)
- ◆ [hw_close](#)
- ◆ [hw_connect](#)
- ◆ [hw_connection_info](#)
- ◆ [hw_cp](#)
- ◆ [hw_deleteobject](#)
- ◆ [hw_docbyanchor](#)
- ◆ [hw_docbyanchorobj](#)
- ◆ [hw_document_attributes](#)
- ◆ [hw_document_bodytag](#)
- ◆ [hw_document_content](#)
- ◆ [hw_document_setcontent](#)
- ◆ [hw_document_size](#)
- ◆ [hw_dummy](#)
- ◆ [hw_edittest](#)
- ◆ [hw_error](#)
- ◆ [hw_errormsg](#)
- ◆ [hw_free_document](#)
- ◆ [hw_getanchors](#)
- ◆ [hw_getanchorsobj](#)
- ◆ [hw_getandlock](#)
- ◆ [hw_getchildcoll](#)
- ◆ [hw_getchildcollobj](#)
- ◆ [hw_getchilddoccoll](#)
- ◆ [hw_getchilddoccollobj](#)
- ◆ [hw_getobject](#)
- ◆ [hw_getobjectbyquery](#)
- ◆ [hw_getobjectbyquerycoll](#)
- ◆ [hw_getobjectbyquerycollobj](#)
- ◆ [hw_getobjectbyqueryobj](#)
- ◆ [hw_getparents](#)

- ◆ [hw_getparentsobj](#)
- ◆ [hw_getrellink](#)
- ◆ [hw_getremote](#)
- ◆ [hw_getremotechildren](#)
- ◆ [hw_getsrcbydestobj](#)
- ◆ [hw_gettext](#)
- ◆ [hw_getusername](#)
- ◆ [hw_identify](#)
- ◆ [hw_incollections](#)
- ◆ [hw_info](#)
- ◆ [hw_inscoll](#)
- ◆ [hw_insdoc](#)
- ◆ [hw_insertanchors](#)
- ◆ [hw_insertdocument](#)
- ◆ [hw_insertobject](#)
- ◆ [hw_mapid](#)
- ◆ [hw_modifyobject](#)
- ◆ [hw_mv](#)
- ◆ [hw_new_document](#)
- ◆ [hw_objrec2array](#)
- ◆ [hw_output_document](#)
- ◆ [hw_pconnect](#)
- ◆ [hw_pipedocument](#)
- ◆ [hw_root](#)
- ◆ [hw_setlinkroot](#)
- ◆ [hw_stat](#)
- ◆ [hw_unlock](#)
- ◆ [hw_who](#)
- ◆ [hwapi_hgcsp](#)
- ◆ [hypot](#)
- |
- - ◆ [ibase_add_user](#)
 - ◆ [ibase_affected_rows](#)
 - ◆ [ibase_backup](#)
 - ◆ [ibase_blob_add](#)
 - ◆ [ibase_blob_cancel](#)
 - ◆ [ibase_blob_close](#)
 - ◆ [ibase_blob_create](#)
 - ◆ [ibase_blob_echo](#)
 - ◆ [ibase_blob_get](#)
 - ◆ [ibase_blob_import](#)
 - ◆ [ibase_blob_info](#)
 - ◆ [ibase_blob_open](#)
 - ◆ [ibase_close](#)
 - ◆ [ibase_commit](#)
 - ◆ [ibase_commit_ret](#)
 - ◆ [ibase_connect](#)
 - ◆ [ibase_db_info](#)
 - ◆ [ibase_delete_user](#)
 - ◆ [ibase_drop_db](#)
 - ◆ [ibase_errcode](#)
 - ◆ [ibase_errmsg](#)
 - ◆ [ibase_execute](#)
 - ◆ [ibase_fetch_assoc](#)

- ◆ [ibase_fetch_object](#)
- ◆ [ibase_fetch_row](#)
- ◆ [ibase_field_info](#)
- ◆ [ibase_free_event_handler](#)
- ◆ [ibase_free_query](#)
- ◆ [ibase_free_result](#)
- ◆ [ibase_gen_id](#)
- ◆ [ibase_maintain_db](#)
- ◆ [ibase_modify_user](#)
- ◆ [ibase_name_result](#)
- ◆ [ibase_num_fields](#)
- ◆ [ibase_num_params](#)
- ◆ [ibase_param_info](#)
- ◆ [ibase_pconnect](#)
- ◆ [ibase_prepare](#)
- ◆ [ibase_query](#)
- ◆ [ibase_restore](#)
- ◆ [ibase_rollback](#)
- ◆ [ibase_rollback_ret](#)
- ◆ [ibase_server_info](#)
- ◆ [ibase_service_attach](#)
- ◆ [ibase_service_detach](#)
- ◆ [ibase_set_event_handler](#)
- ◆ [ibase_timefmt](#)
- ◆ [ibase_trans](#)
- ◆ [ibase_wait_event](#)
- ◆ [icap_close](#)
- ◆ [icap_create_calendar](#)
- ◆ [icap_delete_calendar](#)
- ◆ [icap_delete_event](#)
- ◆ [icap_fetch_event](#)
- ◆ [icap_list_alarms](#)
- ◆ [icap_list_events](#)
- ◆ [icap_open](#)
- ◆ [icap_rename_calendar](#)
- ◆ [icap_reopen](#)
- ◆ [icap_snooze](#)
- ◆ [icap_store_event](#)
- ◆ [iconv](#)
- ◆ [iconv_get_encoding](#)
- ◆ [iconv_mime_decode](#)
- ◆ [iconv_mime_decode_headers](#)
- ◆ [iconv_mime_encode](#)
- ◆ [iconv_set_encoding](#)
- ◆ [iconv_strlen](#)
- ◆ [iconv_strpos](#)
- ◆ [iconv_strrpos](#)
- ◆ [iconv_substr](#)
- ◆ [id3_get_frame_long_name](#)
- ◆ [id3_get_frame_short_name](#)
- ◆ [id3_get_genre_id](#)
- ◆ [id3_get_genre_list](#)
- ◆ [id3_get_genre_name](#)
- ◆ [id3_get_tag](#)

- ◆ [id3_get_version](#)
- ◆ [id3_remove_tag](#)
- ◆ [id3_set_tag](#)
- ◆ [idate](#)
- ◆ [ifx_affected_rows](#)
- ◆ [ifx_blobinfile_mode](#)
- ◆ [ifx_byteasvarchar](#)
- ◆ [ifx_close](#)
- ◆ [ifx_connect](#)
- ◆ [ifx_copy_blob](#)
- ◆ [ifx_create_blob](#)
- ◆ [ifx_create_char](#)
- ◆ [ifx_do](#)
- ◆ [ifx_error](#)
- ◆ [ifx_errormsg](#)
- ◆ [ifx_fetch_row](#)
- ◆ [ifx_fieldproperties](#)
- ◆ [ifx_fieldtypes](#)
- ◆ [ifx_free_blob](#)
- ◆ [ifx_free_char](#)
- ◆ [ifx_free_result](#)
- ◆ [ifx_get_blob](#)
- ◆ [ifx_get_char](#)
- ◆ [ifx_getsqlca](#)
- ◆ [ifx_htmltbl_result](#)
- ◆ [ifx_nullformat](#)
- ◆ [ifx_num_fields](#)
- ◆ [ifx_num_rows](#)
- ◆ [ifx_pconnect](#)
- ◆ [ifx_prepare](#)
- ◆ [ifx_query](#)
- ◆ [ifx_textasvarchar](#)
- ◆ [ifx_update_blob](#)
- ◆ [ifx_update_char](#)
- ◆ [ifxus_close_slob](#)
- ◆ [ifxus_create_slob](#)
- ◆ [ifxus_free_slob](#)
- ◆ [ifxus_open_slob](#)
- ◆ [ifxus_read_slob](#)
- ◆ [ifxus_seek_slob](#)
- ◆ [ifxus_tell_slob](#)
- ◆ [ifxus_write_slob](#)
- ◆ [ignore_user_abort](#)
- ◆ [iis_add_server](#)
- ◆ [iis_get_dir_security](#)
- ◆ [iis_get_script_map](#)
- ◆ [iis_get_server_by_comment](#)
- ◆ [iis_get_server_by_path](#)
- ◆ [iis_get_server_rights](#)
- ◆ [iis_get_service_state](#)
- ◆ [iis_remove_server](#)
- ◆ [iis_set_app_settings](#)
- ◆ [iis_set_dir_security](#)
- ◆ [iis_set_script_map](#)

- ◆ [iis_set_server_rights](#)
- ◆ [iis_start_server](#)
- ◆ [iis_start_service](#)
- ◆ [iis_stop_server](#)
- ◆ [iis_stop_service](#)
- ◆ [image2wbmp](#)
- ◆ [image_type_to_extension](#)
- ◆ [image_type_to_mime_type](#)
- ◆ [imagealphablending](#)
- ◆ [imageantialias](#)
- ◆ [imagearc](#)
- ◆ [imagechar](#)
- ◆ [imagecharup](#)
- ◆ [imagecolorallocate](#)
- ◆ [imagecolorallocatealpha](#)
- ◆ [imagecolorat](#)
- ◆ [imagecolorclosest](#)
- ◆ [imagecolorclosestalpha](#)
- ◆ [imagecolorclosesthw](#)
- ◆ [imagecolordeallocate](#)
- ◆ [imagecolorexact](#)
- ◆ [imagecolorexactalpha](#)
- ◆ [imagecolormatch](#)
- ◆ [imagecolorresolve](#)
- ◆ [imagecolorresolvealpha](#)
- ◆ [imagecolorset](#)
- ◆ [imagecolorsforindex](#)
- ◆ [imagecolorstotal](#)
- ◆ [imagecolortransparent](#)
- ◆ [imageconvolution](#)
- ◆ [imagecopy](#)
- ◆ [imagecopymerge](#)
- ◆ [imagecopymergegray](#)
- ◆ [imagecopyresampled](#)
- ◆ [imagecopyresized](#)
- ◆ [imagecreate](#)
- ◆ [imagecreatefromgd](#)
- ◆ [imagecreatefromgd2](#)
- ◆ [imagecreatefromgd2part](#)
- ◆ [imagecreatefromgif](#)
- ◆ [imagecreatefromjpeg](#)
- ◆ [imagecreatefrompng](#)
- ◆ [imagecreatefromstring](#)
- ◆ [imagecreatefromwbmp](#)
- ◆ [imagecreatefromxbm](#)
- ◆ [imagecreatefromxpm](#)
- ◆ [imagecreatetruecolor](#)
- ◆ [imagedashedline](#)
- ◆ [imagedestroy](#)
- ◆ [imageellipse](#)
- ◆ [imagefill](#)
- ◆ [imagefilledarc](#)
- ◆ [imagefilledellipse](#)
- ◆ [imagefilledpolygon](#)

- ◆ [imagefilledrectangle](#)
- ◆ [imagefilltoborder](#)
- ◆ [imagefilter](#)
- ◆ [imagefontheight](#)
- ◆ [imagefontwidth](#)
- ◆ [imageftbbox](#)
- ◆ [imagefttext](#)
- ◆ [imagegammacorrect](#)
- ◆ [imagegd](#)
- ◆ [imagegd2](#)
- ◆ [imagegif](#)
- ◆ [imageinterlace](#)
- ◆ [imageistruecolor](#)
- ◆ [imagejpeg](#)
- ◆ [imagelayereffect](#)
- ◆ [imageline](#)
- ◆ [imageloadfont](#)
- ◆ [imagepalettecopy](#)
- ◆ [imagepng](#)
- ◆ [imagepolygon](#)
- ◆ [imagepsbbox](#)
- ◆ [imagepscropyfont](#)
- ◆ [imagepsencodefont](#)
- ◆ [imagepsextendfont](#)
- ◆ [imagepsfreefont](#)
- ◆ [imagepsloadfont](#)
- ◆ [imagepslantfont](#)
- ◆ [imagepstext](#)
- ◆ [imagerectangle](#)
- ◆ [imagerotate](#)
- ◆ [imagesavealpha](#)
- ◆ [imagesetbrush](#)
- ◆ [imagesetpixel](#)
- ◆ [imagesetstyle](#)
- ◆ [imagesetthickness](#)
- ◆ [imagesettile](#)
- ◆ [imagestring](#)
- ◆ [imagestringup](#)
- ◆ [imagesx](#)
- ◆ [imagesy](#)
- ◆ [imagetruecolorpalette](#)
- ◆ [imagettfbbox](#)
- ◆ [imagettftext](#)
- ◆ [imagetypes](#)
- ◆ [imagewbmp](#)
- ◆ [imagexbm](#)
- ◆ [imap_8bit](#)
- ◆ [imap_alerts](#)
- ◆ [imap_append](#)
- ◆ [imap_base64](#)
- ◆ [imap_binary](#)
- ◆ [imap_body](#)
- ◆ [imap_bodystruct](#)
- ◆ [imap_check](#)

- ◆ [imap_clearflag_full](#)
- ◆ [imap_close](#)
- ◆ [imap_createmailbox](#)
- ◆ [imap_delete](#)
- ◆ [imap_deletemailbox](#)
- ◆ [imap_errors](#)
- ◆ [imap_expunge](#)
- ◆ [imap_fetch_overview](#)
- ◆ [imap_fetchbody](#)
- ◆ [imap_fetchheader](#)
- ◆ [imap_fetchstructure](#)
- ◆ [imap_get_quota](#)
- ◆ [imap_get_quotaroot](#)
- ◆ [imap_getacl](#)
- ◆ [imap_getmailboxes](#)
- ◆ [imap_getsubscribed](#)
- ◆ [imap_header](#)
- ◆ [imap_headerinfo](#)
- ◆ [imap_headers](#)
- ◆ [imap_last_error](#)
- ◆ [imap_list](#)
- ◆ [imap_listmailbox](#)
- ◆ [imap_listscan](#)
- ◆ [imap_listsubscribed](#)
- ◆ [imap_lsub](#)
- ◆ [imap_mail](#)
- ◆ [imap_mail_compose](#)
- ◆ [imap_mail_copy](#)
- ◆ [imap_mail_move](#)
- ◆ [imap_mailboxmsginfo](#)
- ◆ [imap_mime_header_decode](#)
- ◆ [imap_msgno](#)
- ◆ [imap_num_msg](#)
- ◆ [imap_num_recent](#)
- ◆ [imap_open](#)
- ◆ [imap_ping](#)
- ◆ [imap_qprint](#)
- ◆ [imap_renamemailbox](#)
- ◆ [imap_reopen](#)
- ◆ [imap_rfc822_parse_adrlist](#)
- ◆ [imap_rfc822_parse_headers](#)
- ◆ [imap_rfc822_write_address](#)
- ◆ [imap_scanmailbox](#)
- ◆ [imap_search](#)
- ◆ [imap_set_quota](#)
- ◆ [imap_setacl](#)
- ◆ [imap_setflag_full](#)
- ◆ [imap_sort](#)
- ◆ [imap_status](#)
- ◆ [imap_subscribe](#)
- ◆ [imap_thread](#)
- ◆ [imap_timeout](#)
- ◆ [imap_uid](#)
- ◆ [imap_undelete](#)

- ◆ [imap_unsubscribe](#)
- ◆ [imap_utf7_decode](#)
- ◆ [imap_utf7_encode](#)
- ◆ [imap_utf8](#)
- ◆ [implode](#)
- ◆ [import_request_variables](#)
- ◆ [in_array](#)
- ◆ [inet_ntop](#)
- ◆ [inet_pton](#)
- ◆ [ingres_autocommit](#)
- ◆ [ingres_close](#)
- ◆ [ingres_commit](#)
- ◆ [ingres_connect](#)
- ◆ [ingres_cursor](#)
- ◆ [ingres_errno](#)
- ◆ [ingres_error](#)
- ◆ [ingres_errsqlstate](#)
- ◆ [ingres_fetch_array](#)
- ◆ [ingres_fetch_object](#)
- ◆ [ingres_fetch_row](#)
- ◆ [ingres_field_length](#)
- ◆ [ingres_field_name](#)
- ◆ [ingres_field_nullable](#)
- ◆ [ingres_field_precision](#)
- ◆ [ingres_field_scale](#)
- ◆ [ingres_field_type](#)
- ◆ [ingres_num_fields](#)
- ◆ [ingres_num_rows](#)
- ◆ [ingres_pconnect](#)
- ◆ [ingres_query](#)
- ◆ [ingres_rollback](#)
- ◆ [ini_alter](#)
- ◆ [ini_get](#)
- ◆ [ini_get_all](#)
- ◆ [ini_restore](#)
- ◆ [ini_set](#)
- ◆ [interface_exists](#)
- ◆ [intval](#)
- ◆ [ip2long](#)
- ◆ [iptcembed](#)
- ◆ [iptcparse](#)
- ◆ [ircg_channel_mode](#)
- ◆ [ircg_disconnect](#)
- ◆ [ircg_eval_ecmascript_params](#)
- ◆ [ircg_fetch_error_msg](#)
- ◆ [ircg_get_username](#)
- ◆ [ircg_html_encode](#)
- ◆ [ircg_ignore_add](#)
- ◆ [ircg_ignore_del](#)
- ◆ [ircg_invite](#)
- ◆ [ircg_is_conn_alive](#)
- ◆ [ircg_join](#)
- ◆ [ircg_kick](#)
- ◆ [ircg_list](#)

- ◆ [ircg_lookup_format_messages](#)
 - ◆ [ircg_lusers](#)
 - ◆ [ircg_msg](#)
 - ◆ [ircg_names](#)
 - ◆ [ircg_nick](#)
 - ◆ [ircg_nickname_escape](#)
 - ◆ [ircg_nickname_unescape](#)
 - ◆ [ircg_notice](#)
 - ◆ [ircg_oper](#)
 - ◆ [ircg_part](#)
 - ◆ [ircg_pconnect](#)
 - ◆ [ircg_register_format_messages](#)
 - ◆ [ircg_set_current](#)
 - ◆ [ircg_set_file](#)
 - ◆ [ircg_set_on_die](#)
 - ◆ [ircg_topic](#)
 - ◆ [ircg_who](#)
 - ◆ [ircg_whois](#)
 - ◆ [is_a](#)
 - ◆ [is_array](#)
 - ◆ [is_bool](#)
 - ◆ [is_callable](#)
 - ◆ [is_dir](#)
 - ◆ [is_double](#)
 - ◆ [is_executable](#)
 - ◆ [is_file](#)
 - ◆ [is_finite](#)
 - ◆ [is_float](#)
 - ◆ [is_infinite](#)
 - ◆ [is_int](#)
 - ◆ [is_integer](#)
 - ◆ [is_link](#)
 - ◆ [is_long](#)
 - ◆ [is_nan](#)
 - ◆ [is_null](#)
 - ◆ [is_numeric](#)
 - ◆ [is_object](#)
 - ◆ [is_readable](#)
 - ◆ [is_real](#)
 - ◆ [is_resource](#)
 - ◆ [is_scalar](#)
 - ◆ [is_soap_fault](#)
 - ◆ [is_string](#)
 - ◆ [is_subclass_of](#)
 - ◆ [is_uploaded_file](#)
 - ◆ [is_writable](#)
 - ◆ [is_writeable](#)
 - ◆ [isset](#)
 - ◆ [iterator_count](#)
 - ◆ [iterator_to_array](#)
- J
- ◆ [java_last_exception_clear](#)
 - ◆ [java_last_exception_get](#)
 - ◆ [jddayofweek](#)

- ◆ [jdmonthname](#)
- ◆ [jdtofrench](#)
- ◆ [jdtogregorian](#)
- ◆ [jdtojewish](#)
- ◆ [jdtojulian](#)
- ◆ [jdtonix](#)
- ◆ [jewishtojd](#)
- ◆ [join](#)
- ◆ [jpeg2wbmp](#)
- ◆ [juliantojd](#)
- **K**
- - ◆ [kadm5_chpass_principal](#)
 - ◆ [kadm5_create_principal](#)
 - ◆ [kadm5_delete_principal](#)
 - ◆ [kadm5_destroy](#)
 - ◆ [kadm5_flush](#)
 - ◆ [kadm5_get_policies](#)
 - ◆ [kadm5_get_principal](#)
 - ◆ [kadm5_get_principals](#)
 - ◆ [kadm5_init_with_password](#)
 - ◆ [kadm5_modify_principal](#)
 - ◆ [key](#)
 - ◆ [krsort](#)
 - ◆ [ksort](#)
- **L**
- - ◆ [lcg_value](#)
 - ◆ [ldap_8859_to_t61](#)
 - ◆ [ldap_add](#)
 - ◆ [ldap_bind](#)
 - ◆ [ldap_close](#)
 - ◆ [ldap_compare](#)
 - ◆ [ldap_connect](#)
 - ◆ [ldap_count_entries](#)
 - ◆ [ldap_delete](#)
 - ◆ [ldap_dn2ufn](#)
 - ◆ [ldap_err2str](#)
 - ◆ [ldap_errno](#)
 - ◆ [ldap_error](#)
 - ◆ [ldap_explode_dn](#)
 - ◆ [ldap_first_attribute](#)
 - ◆ [ldap_first_entry](#)
 - ◆ [ldap_first_reference](#)
 - ◆ [ldap_free_result](#)
 - ◆ [ldap_get_attributes](#)
 - ◆ [ldap_get_dn](#)
 - ◆ [ldap_get_entries](#)
 - ◆ [ldap_get_option](#)
 - ◆ [ldap_get_values](#)
 - ◆ [ldap_get_values_len](#)
 - ◆ [ldap_list](#)
 - ◆ [ldap_mod_add](#)
 - ◆ [ldap_mod_del](#)
 - ◆ [ldap_mod_replace](#)
 - ◆ [ldap_modify](#)

- ◆ [ldap_next_attribute](#)
- ◆ [ldap_next_entry](#)
- ◆ [ldap_next_reference](#)
- ◆ [ldap_parse_reference](#)
- ◆ [ldap_parse_result](#)
- ◆ [ldap_read](#)
- ◆ [ldap_rename](#)
- ◆ [ldap_sasl_bind](#)
- ◆ [ldap_search](#)
- ◆ [ldap_set_option](#)
- ◆ [ldap_set_rebind_proc](#)
- ◆ [ldap_sort](#)
- ◆ [ldap_start_tls](#)
- ◆ [ldap_t61_to_8859](#)
- ◆ [ldap_unbind](#)
- ◆ [levenshtein](#)
- ◆ [libxml_clear_errors](#)
- ◆ [libxml_get_errors](#)
- ◆ [libxml_get_last_error](#)
- ◆ [libxml_set_streams_context](#)
- ◆ [libxml_use_internal_errors](#)
- ◆ [limititerator::getposition](#)
- ◆ [limititerator::next](#)
- ◆ [limititerator::rewind](#)
- ◆ [limititerator::seek](#)
- ◆ [limititerator::valid](#)
- ◆ [link](#)
- ◆ [linkinfo](#)
- ◆ [list](#)
- ◆ [localeconv](#)
- ◆ [localtime](#)
- ◆ [log](#)
- ◆ [log10](#)
- ◆ [log1p](#)
- ◆ [long2ip](#)
- ◆ [lstat](#)
- ◆ [ltrim](#)
- ◆ [lzf_compress](#)
- ◆ [lzf_decompress](#)
- ◆ [lzf_optimized_for](#)
- **M**
- - ◆ [m_checkstatus](#)
 - ◆ [m_completeauthorizations](#)
 - ◆ [m_connect](#)
 - ◆ [m_connectionerror](#)
 - ◆ [m_deletetrans](#)
 - ◆ [m_destroyconn](#)
 - ◆ [m_destroyengine](#)
 - ◆ [m_getcell](#)
 - ◆ [m_getcellbynum](#)
 - ◆ [m_getcommadelimited](#)
 - ◆ [m_getheader](#)
 - ◆ [m_initconn](#)
 - ◆ [m_initengine](#)

- ◆ [m_iscommadelimited](#)
- ◆ [m_maxconntimeout](#)
- ◆ [m_monitor](#)
- ◆ [m_numcolumns](#)
- ◆ [m_numrows](#)
- ◆ [m_parsecommadelimited](#)
- ◆ [m_responsekeys](#)
- ◆ [m_responseparam](#)
- ◆ [m_returnstatus](#)
- ◆ [m_setblocking](#)
- ◆ [m_setdropfile](#)
- ◆ [m_setip](#)
- ◆ [m_setssl](#)
- ◆ [m_setssl_cafile](#)
- ◆ [m_setssl_files](#)
- ◆ [m_settimeout](#)
- ◆ [m_sslcert_gen_hash](#)
- ◆ [m_transactionssent](#)
- ◆ [m_transinqueue](#)
- ◆ [m_transkeyval](#)
- ◆ [m_transnew](#)
- ◆ [m_transsend](#)
- ◆ [m_uwait](#)
- ◆ [m_validateidentifier](#)
- ◆ [m_verifyconnection](#)
- ◆ [m_verifysslcert](#)
- ◆ [mail](#)
- ◆ [mailparse_determine_best_xfer_encoding](#)
- ◆ [mailparse_msg_create](#)
- ◆ [mailparse_msg_extract_part](#)
- ◆ [mailparse_msg_extract_part_file](#)
- ◆ [mailparse_msg_free](#)
- ◆ [mailparse_msg_get_part](#)
- ◆ [mailparse_msg_get_part_data](#)
- ◆ [mailparse_msg_get_structure](#)
- ◆ [mailparse_msg_parse](#)
- ◆ [mailparse_msg_parse_file](#)
- ◆ [mailparse_rfc822_parse_addresses](#)
- ◆ [mailparse_stream_encode](#)
- ◆ [mailparse_uudecode_all](#)
- ◆ [main](#)
- ◆ [max](#)
- ◆ [maxdb\(\)](#)
- ◆ [maxdb->affected_rows](#)
- ◆ [maxdb->change_user](#)
- ◆ [maxdb->character_set_name](#)
- ◆ [maxdb->close](#)
- ◆ [maxdb->close_long_data](#)
- ◆ [maxdb->commit](#)
- ◆ [maxdb->disable_reads_from_master](#)
- ◆ [maxdb->errno](#)
- ◆ [maxdb->fetch_assoc](#)
- ◆ [maxdb->field_count](#)
- ◆ [maxdb->get_host_info](#)

- ◆ [maxdb->info](#)
- ◆ [maxdb->insert_id](#)
- ◆ [maxdb->kill](#)
- ◆ [maxdb->more_results](#)
- ◆ [maxdb->multi_query](#)
- ◆ [maxdb->next_result](#)
- ◆ [maxdb->options](#)
- ◆ [maxdb->ping](#)
- ◆ [maxdb->prepare](#)
- ◆ [maxdb->protocol_version](#)
- ◆ [maxdb->query](#)
- ◆ [maxdb->real_connect](#)
- ◆ [maxdb->real_escape_string](#)
- ◆ [maxdb->real_query](#)
- ◆ [maxdb->rollback](#)
- ◆ [maxdb->rpl_query_type](#)
- ◆ [maxdb->select_db](#)
- ◆ [maxdb->send_query](#)
- ◆ [maxdb->server_info](#)
- ◆ [maxdb->sqlstate](#)
- ◆ [maxdb->ssl_set](#)
- ◆ [maxdb->stat](#)
- ◆ [maxdb->stmt_init](#)
- ◆ [maxdb->store_result](#)
- ◆ [maxdb->store_result](#)
- ◆ [maxdb->thread_id](#)
- ◆ [maxdb->use_result](#)
- ◆ [maxdb->warning_count](#)
- ◆ [maxdb_bind_param](#)
- ◆ [maxdb_bind_result](#)
- ◆ [maxdb_client_encoding](#)
- ◆ [maxdb_connect_errno](#)
- ◆ [maxdb_connect_error](#)
- ◆ [maxdb_debug](#)
- ◆ [maxdb_disable_rpl_parse](#)
- ◆ [maxdb_dump_debug_info](#)
- ◆ [maxdb_embedded_connect](#)
- ◆ [maxdb_enable_reads_from_master](#)
- ◆ [maxdb_enable_rpl_parse](#)
- ◆ [maxdb_error](#)
- ◆ [maxdb_escape_string](#)
- ◆ [maxdb_execute](#)
- ◆ [maxdb_fetch](#)
- ◆ [maxdb_get_client_info](#)
- ◆ [maxdb_get_client_version](#)
- ◆ [maxdb_get_metadata](#)
- ◆ [maxdb_get_server_version](#)
- ◆ [maxdb_init](#)
- ◆ [maxdb_master_query](#)
- ◆ [maxdb_num_rows](#)
- ◆ [maxdb_param_count](#)
- ◆ [maxdb_report](#)
- ◆ [maxdb_rpl_parse_enabled](#)
- ◆ [maxdb_rpl_probe](#)

- ◆ [maxdb_send_long_data](#)
- ◆ [maxdb_server_end](#)
- ◆ [maxdb_server_init](#)
- ◆ [maxdb_set_opt](#)
- ◆ [maxdb_stmt->affected_rows](#)
- ◆ [maxdb_stmt->close](#)
- ◆ [maxdb_stmt->errno](#)
- ◆ [maxdb_stmt->error](#)
- ◆ [maxdb_stmt_result_metadata](#)
- ◆ [maxdb_stmt_sqlstate](#)
- ◆ [maxdb_thread_safe](#)
- ◆ [mb_convert_case](#)
- ◆ [mb_convert_encoding](#)
- ◆ [mb_convert_kana](#)
- ◆ [mb_convert_variables](#)
- ◆ [mb_decode_mimeheader](#)
- ◆ [mb_decode_numericentity](#)
- ◆ [mb_detect_encoding](#)
- ◆ [mb_detect_order](#)
- ◆ [mb_encode_mimeheader](#)
- ◆ [mb_encode_numericentity](#)
- ◆ [mb_ereg](#)
- ◆ [mb_ereg_match](#)
- ◆ [mb_ereg_replace](#)
- ◆ [mb_ereg_search](#)
- ◆ [mb_ereg_search_getpos](#)
- ◆ [mb_ereg_search_getregs](#)
- ◆ [mb_ereg_search_init](#)
- ◆ [mb_ereg_search_pos](#)
- ◆ [mb_ereg_search_regs](#)
- ◆ [mb_ereg_search_setpos](#)
- ◆ [mb_eregi](#)
- ◆ [mb_eregi_replace](#)
- ◆ [mb_get_info](#)
- ◆ [mb_http_input](#)
- ◆ [mb_http_output](#)
- ◆ [mb_internal_encoding](#)
- ◆ [mb_language](#)
- ◆ [mb_list_encodings](#)
- ◆ [mb_output_handler](#)
- ◆ [mb_parse_str](#)
- ◆ [mb_preferred_mime_name](#)
- ◆ [mb_regex_encoding](#)
- ◆ [mb_regex_set_options](#)
- ◆ [mb_send_mail](#)
- ◆ [mb_split](#)
- ◆ [mb_strcut](#)
- ◆ [mb_striwidth](#)
- ◆ [mb_strlen](#)
- ◆ [mb_strpos](#)
- ◆ [mb_strrpos](#)
- ◆ [mb_strtolower](#)
- ◆ [mb_strtoupper](#)
- ◆ [mb_strwidth](#)

- ◆ [mb_substitute_character](#)
- ◆ [mb_substr](#)
- ◆ [mb_substr_count](#)
- ◆ [mcal_append_event](#)
- ◆ [mcal_close](#)
- ◆ [mcal_create_calendar](#)
- ◆ [mcal_date_compare](#)
- ◆ [mcal_date_valid](#)
- ◆ [mcal_day_of_week](#)
- ◆ [mcal_day_of_year](#)
- ◆ [mcal_days_in_month](#)
- ◆ [mcal_delete_calendar](#)
- ◆ [mcal_delete_event](#)
- ◆ [mcal_event_add_attribute](#)
- ◆ [mcal_event_init](#)
- ◆ [mcal_event_set_alarm](#)
- ◆ [mcal_event_set_category](#)
- ◆ [mcal_event_set_class](#)
- ◆ [mcal_event_set_description](#)
- ◆ [mcal_event_set_end](#)
- ◆ [mcal_event_set_recur_daily](#)
- ◆ [mcal_event_set_recur_monthly_mday](#)
- ◆ [mcal_event_set_recur_monthly_wday](#)
- ◆ [mcal_event_set_recur_none](#)
- ◆ [mcal_event_set_recur_weekly](#)
- ◆ [mcal_event_set_recur_yearly](#)
- ◆ [mcal_event_set_start](#)
- ◆ [mcal_event_set_title](#)
- ◆ [mcal_expunge](#)
- ◆ [mcal_fetch_current_stream_event](#)
- ◆ [mcal_fetch_event](#)
- ◆ [mcal_is_leap_year](#)
- ◆ [mcal_list_alarms](#)
- ◆ [mcal_list_events](#)
- ◆ [mcal_next_recurrence](#)
- ◆ [mcal_open](#)
- ◆ [mcal_popen](#)
- ◆ [mcal_rename_calendar](#)
- ◆ [mcal_reopen](#)
- ◆ [mcal_snooze](#)
- ◆ [mcal_store_event](#)
- ◆ [mcal_time_valid](#)
- ◆ [mcal_week_of_year](#)
- ◆ [mcrypt_cbc](#)
- ◆ [mcrypt_cfb](#)
- ◆ [mcrypt_create_iv](#)
- ◆ [mcrypt_decrypt](#)
- ◆ [mcrypt_ecb](#)
- ◆ [mcrypt_enc_get_algorithms_name](#)
- ◆ [mcrypt_enc_get_block_size](#)
- ◆ [mcrypt_enc_get_iv_size](#)
- ◆ [mcrypt_enc_get_key_size](#)
- ◆ [mcrypt_enc_get_modes_name](#)
- ◆ [mcrypt_enc_get_supported_key_sizes](#)

- ◆ [mdecrypt_enc is block algorithm](#)
- ◆ [mdecrypt_enc is block algorithm mode](#)
- ◆ [mdecrypt_enc is block mode](#)
- ◆ [mdecrypt_enc self test](#)
- ◆ [mdecrypt_encrypt](#)
- ◆ [mdecrypt_generic](#)
- ◆ [mdecrypt_generic deinit](#)
- ◆ [mdecrypt_generic_end](#)
- ◆ [mdecrypt_generic_init](#)
- ◆ [mdecrypt_get_block_size](#)
- ◆ [mdecrypt_get_cipher_name](#)
- ◆ [mdecrypt_get_iv_size](#)
- ◆ [mdecrypt_get_key_size](#)
- ◆ [mdecrypt_list_algorithms](#)
- ◆ [mdecrypt_list_modes](#)
- ◆ [mdecrypt_module_close](#)
- ◆ [mdecrypt_module_get_algo_block_size](#)
- ◆ [mdecrypt_module_get_algo_key_size](#)
- ◆ [mdecrypt_module_get_supported_key_sizes](#)
- ◆ [mdecrypt_module_is_block_algorithm](#)
- ◆ [mdecrypt_module_is_block_algorithm_mode](#)
- ◆ [mdecrypt_module_is_block_mode](#)
- ◆ [mdecrypt_module_open](#)
- ◆ [mdecrypt_module_self_test](#)
- ◆ [mdecrypt_ofb](#)
- ◆ [md5](#)
- ◆ [md5_file](#)
- ◆ [mdecrypt_generic](#)
- ◆ [memcache::add](#)
- ◆ [memcache::close](#)
- ◆ [memcache::connect](#)
- ◆ [memcache::decrement](#)
- ◆ [memcache::delete](#)
- ◆ [memcache::flush](#)
- ◆ [memcache::get](#)
- ◆ [memcache::getstats](#)
- ◆ [memcache::getversion](#)
- ◆ [memcache::increment](#)
- ◆ [memcache::pconnect](#)
- ◆ [memcache::replace](#)
- ◆ [memcache::set](#)
- ◆ [memcache_debug](#)
- ◆ [memory_get_usage](#)
- ◆ [metaphone](#)
- ◆ [method_exists](#)
- ◆ [mhash](#)
- ◆ [mhash_count](#)
- ◆ [mhash_get_block_size](#)
- ◆ [mhash_get_hash_name](#)
- ◆ [mhash_keygen_s2k](#)
- ◆ [microtime](#)
- ◆ [min](#)
- ◆ [ming_keypress](#)
- ◆ [ming_setcubicthreshold](#)

- ◆ [ming_setscale](#)
- ◆ [ming_useconstants](#)
- ◆ [ming_useswfversion](#)
- ◆ [mkdir](#)
- ◆ [mktime](#)
- ◆ [money_format](#)
- ◆ [move_uploaded_file](#)
- ◆ [msession_connect](#)
- ◆ [msession_count](#)
- ◆ [msession_create](#)
- ◆ [msession_destroy](#)
- ◆ [msession_disconnect](#)
- ◆ [msession_find](#)
- ◆ [msession_get](#)
- ◆ [msession_get_array](#)
- ◆ [msession_get_data](#)
- ◆ [msession_inc](#)
- ◆ [msession_list](#)
- ◆ [msession_listvar](#)
- ◆ [msession_lock](#)
- ◆ [msession_plugin](#)
- ◆ [msession_randstr](#)
- ◆ [msession_set](#)
- ◆ [msession_set_array](#)
- ◆ [msession_set_data](#)
- ◆ [msession_timeout](#)
- ◆ [msession_uniq](#)
- ◆ [msession_unlock](#)
- ◆ [msg_get_queue](#)
- ◆ [msg_receive](#)
- ◆ [msg_remove_queue](#)
- ◆ [msg_send](#)
- ◆ [msg_set_queue](#)
- ◆ [msg_stat_queue](#)
- ◆ [mysql](#)
- ◆ [mysql_affected_rows](#)
- ◆ [mysql_close](#)
- ◆ [mysql_connect](#)
- ◆ [mysql_create_db](#)
- ◆ [mysql_createdb](#)
- ◆ [mysql_data_seek](#)
- ◆ [mysql_db_query](#)
- ◆ [mysql_dbname](#)
- ◆ [mysql_drop_db](#)
- ◆ [mysql_error](#)
- ◆ [mysql_fetch_array](#)
- ◆ [mysql_fetch_field](#)
- ◆ [mysql_fetch_object](#)
- ◆ [mysql_fetch_row](#)
- ◆ [mysql_field_flags](#)
- ◆ [mysql_field_len](#)
- ◆ [mysql_field_name](#)
- ◆ [mysql_field_seek](#)
- ◆ [mysql_field_table](#)

- ◆ [mysql_field_type](#)
- ◆ [mysql_fieldflags](#)
- ◆ [mysql_fieldlen](#)
- ◆ [mysql_fieldname](#)
- ◆ [mysql_fieldtable](#)
- ◆ [mysql_fieldtype](#)
- ◆ [mysql_free_result](#)
- ◆ [mysql_list_dbs](#)
- ◆ [mysql_list_fields](#)
- ◆ [mysql_list_tables](#)
- ◆ [mysql_num_fields](#)
- ◆ [mysql_num_rows](#)
- ◆ [mysql_numfields](#)
- ◆ [mysql_numrows](#)
- ◆ [mysql_pconnect](#)
- ◆ [mysql_query](#)
- ◆ [mysql_regcase](#)
- ◆ [mysql_result](#)
- ◆ [mysql_select_db](#)
- ◆ [mysql_tablename](#)
- ◆ [mssql_bind](#)
- ◆ [mssql_close](#)
- ◆ [mssql_connect](#)
- ◆ [mssql_data_seek](#)
- ◆ [mssql_execute](#)
- ◆ [mssql_fetch_array](#)
- ◆ [mssql_fetch_assoc](#)
- ◆ [mssql_fetch_batch](#)
- ◆ [mssql_fetch_field](#)
- ◆ [mssql_fetch_object](#)
- ◆ [mssql_fetch_row](#)
- ◆ [mssql_field_length](#)
- ◆ [mssql_field_name](#)
- ◆ [mssql_field_seek](#)
- ◆ [mssql_field_type](#)
- ◆ [mssql_free_result](#)
- ◆ [mssql_free_statement](#)
- ◆ [mssql_get_last_message](#)
- ◆ [mssql_guid_string](#)
- ◆ [mssql_init](#)
- ◆ [mssql_min_error_severity](#)
- ◆ [mssql_min_message_severity](#)
- ◆ [mssql_next_result](#)
- ◆ [mssql_num_fields](#)
- ◆ [mssql_num_rows](#)
- ◆ [mssql_pconnect](#)
- ◆ [mssql_query](#)
- ◆ [mssql_result](#)
- ◆ [mssql_rows_affected](#)
- ◆ [mssql_select_db](#)
- ◆ [mt_getrandmax](#)
- ◆ [mt_rand](#)
- ◆ [mt_srand](#)
- ◆ [muscat_close](#)

- ◆ [muscat_get](#)
- ◆ [muscat_give](#)
- ◆ [muscat_setup](#)
- ◆ [muscat_setup_net](#)
- ◆ [mysql_affected_rows](#)
- ◆ [mysql_change_user](#)
- ◆ [mysql_client_encoding](#)
- ◆ [mysql_close](#)
- ◆ [mysql_connect](#)
- ◆ [mysql_create_db](#)
- ◆ [mysql_data_seek](#)
- ◆ [mysql_db_name](#)
- ◆ [mysql_db_query](#)
- ◆ [mysql_drop_db](#)
- ◆ [mysql_errno](#)
- ◆ [mysql_error](#)
- ◆ [mysql_escape_string](#)
- ◆ [mysql_fetch_array](#)
- ◆ [mysql_fetch_assoc](#)
- ◆ [mysql_fetch_field](#)
- ◆ [mysql_fetch_lengths](#)
- ◆ [mysql_fetch_object](#)
- ◆ [mysql_fetch_row](#)
- ◆ [mysql_field_flags](#)
- ◆ [mysql_field_len](#)
- ◆ [mysql_field_name](#)
- ◆ [mysql_field_seek](#)
- ◆ [mysql_field_table](#)
- ◆ [mysql_field_type](#)
- ◆ [mysql_free_result](#)
- ◆ [mysql_get_client_info](#)
- ◆ [mysql_get_host_info](#)
- ◆ [mysql_get_proto_info](#)
- ◆ [mysql_get_server_info](#)
- ◆ [mysql_info](#)
- ◆ [mysql_insert_id](#)
- ◆ [mysql_list_dbs](#)
- ◆ [mysql_list_fields](#)
- ◆ [mysql_list_processes](#)
- ◆ [mysql_list_tables](#)
- ◆ [mysql_num_fields](#)
- ◆ [mysql_num_rows](#)
- ◆ [mysql_pconnect](#)
- ◆ [mysql_ping](#)
- ◆ [mysql_query](#)
- ◆ [mysql_real_escape_string](#)
- ◆ [mysql_result](#)
- ◆ [mysql_select_db](#)
- ◆ [mysql_stat](#)
- ◆ [mysql_tablename](#)
- ◆ [mysql_thread_id](#)
- ◆ [mysql_unbuffered_query](#)
- ◆ [mysqli\(\)](#)
- ◆ [mysqli->affected_rows](#)

- ◆ [mysql->autocommit](#)
- ◆ [mysql->change_user](#)
- ◆ [mysql->character_set_name](#)
- ◆ [mysql->close](#)
- ◆ [mysql->commit](#)
- ◆ [mysql->disable_reads_from_master](#)
- ◆ [mysql->dump_debug_info](#)
- ◆ [mysql->errno](#)
- ◆ [mysql->fetch_assoc](#)
- ◆ [mysql->field_count](#)
- ◆ [mysql->get_host_info](#)
- ◆ [mysql->info](#)
- ◆ [mysql->insert_id](#)
- ◆ [mysql->kill](#)
- ◆ [mysql->more_results](#)
- ◆ [mysql->multi_query](#)
- ◆ [mysql->next_result](#)
- ◆ [mysql->options](#)
- ◆ [mysql->ping](#)
- ◆ [mysql->prepare](#)
- ◆ [mysql->protocol_version](#)
- ◆ [mysql->query](#)
- ◆ [mysql->real_connect](#)
- ◆ [mysql->real_escape_string](#)
- ◆ [mysql->real_query](#)
- ◆ [mysql->rollback](#)
- ◆ [mysql->rpl_query_type](#)
- ◆ [mysql->select_db](#)
- ◆ [mysql->send_query](#)
- ◆ [mysql->server_info](#)
- ◆ [mysql->set_charset](#)
- ◆ [mysql->sqlstate](#)
- ◆ [mysql->ssl_set](#)
- ◆ [mysql->stat](#)
- ◆ [mysql->stmt_init](#)
- ◆ [mysql->store_result](#)
- ◆ [mysql->thread_id](#)
- ◆ [mysql->use_result](#)
- ◆ [mysql->warning_count](#)
- ◆ [mysql_bind_param](#)
- ◆ [mysql_bind_result](#)
- ◆ [mysql_client_encoding](#)
- ◆ [mysql_connect_errno](#)
- ◆ [mysql_connect_error](#)
- ◆ [mysql_debug](#)
- ◆ [mysql_disable_rpl_parse](#)
- ◆ [mysql_embedded_connect](#)
- ◆ [mysql_enable_reads_from_master](#)
- ◆ [mysql_enable_rpl_parse](#)
- ◆ [mysql_error](#)
- ◆ [mysql_escape_string](#)
- ◆ [mysql_execute](#)
- ◆ [mysql_fetch](#)
- ◆ [mysql_get_client_info](#)

- ◆ [mysql_get_client_version](#)
- ◆ [mysql_get_metadata](#)
- ◆ [mysql_get_server_version](#)
- ◆ [mysql_init](#)
- ◆ [mysql_master_query](#)
- ◆ [mysql_num_rows](#)
- ◆ [mysql_param_count](#)
- ◆ [mysql_report](#)
- ◆ [mysql_rpl_parse_enabled](#)
- ◆ [mysql_rpl_probe](#)
- ◆ [mysql_send_long_data](#)
- ◆ [mysql_server_end](#)
- ◆ [mysql_server_init](#)
- ◆ [mysql_set_opt](#)
- ◆ [mysql_stmt->affected_rows](#)
- ◆ [mysql_stmt->close](#)
- ◆ [mysql_stmt->errno](#)
- ◆ [mysql_stmt->error](#)
- ◆ [mysql_stmt->store_result](#)
- ◆ [mysql_stmt_result_metadata](#)
- ◆ [mysql_stmt_sqlstate](#)
- ◆ [mysql_thread_safe](#)
- N
- - ◆ [natcasesort](#)
 - ◆ [natsort](#)
 - ◆ [ncurses_addch](#)
 - ◆ [ncurses_addchnstr](#)
 - ◆ [ncurses_addchstr](#)
 - ◆ [ncurses_addnstr](#)
 - ◆ [ncurses_addstr](#)
 - ◆ [ncurses_assume_default_colors](#)
 - ◆ [ncurses_attroff](#)
 - ◆ [ncurses_atron](#)
 - ◆ [ncurses_attrset](#)
 - ◆ [ncurses_baudrate](#)
 - ◆ [ncurses_beep](#)
 - ◆ [ncurses_bkgd](#)
 - ◆ [ncurses_bkgdset](#)
 - ◆ [ncurses_border](#)
 - ◆ [ncurses_bottom_panel](#)
 - ◆ [ncurses_can_change_color](#)
 - ◆ [ncurses_cbreak](#)
 - ◆ [ncurses_clear](#)
 - ◆ [ncurses_clrtobot](#)
 - ◆ [ncurses_clrtoeol](#)
 - ◆ [ncurses_color_content](#)
 - ◆ [ncurses_color_set](#)
 - ◆ [ncurses_curs_set](#)
 - ◆ [ncurses_def_prog_mode](#)
 - ◆ [ncurses_def_shell_mode](#)
 - ◆ [ncurses_define_key](#)
 - ◆ [ncurses_del_panel](#)
 - ◆ [ncurses_delay_output](#)
 - ◆ [ncurses_delch](#)

- ◆ [ncurses_deleteln](#)
- ◆ [ncurses_delwin](#)
- ◆ [ncurses_doupdate](#)
- ◆ [ncurses_echo](#)
- ◆ [ncurses_echochar](#)
- ◆ [ncurses_end](#)
- ◆ [ncurses_erase](#)
- ◆ [ncurses_erasechar](#)
- ◆ [ncurses_filter](#)
- ◆ [ncurses_flash](#)
- ◆ [ncurses_flushinp](#)
- ◆ [ncurses_getch](#)
- ◆ [ncurses_getmaxyx](#)
- ◆ [ncurses_getmouse](#)
- ◆ [ncurses_getyx](#)
- ◆ [ncurses_halfdelay](#)
- ◆ [ncurses_has_colors](#)
- ◆ [ncurses_has_ic](#)
- ◆ [ncurses_has_il](#)
- ◆ [ncurses_has_key](#)
- ◆ [ncurses_hide_panel](#)
- ◆ [ncurses_hline](#)
- ◆ [ncurses_inch](#)
- ◆ [ncurses_init](#)
- ◆ [ncurses_init_color](#)
- ◆ [ncurses_init_pair](#)
- ◆ [ncurses_insch](#)
- ◆ [ncurses_insdelln](#)
- ◆ [ncurses_insertln](#)
- ◆ [ncurses_insstr](#)
- ◆ [ncurses_instr](#)
- ◆ [ncurses_isendwin](#)
- ◆ [ncurses_keyok](#)
- ◆ [ncurses_keypad](#)
- ◆ [ncurses_killchar](#)
- ◆ [ncurses_longname](#)
- ◆ [ncurses_meta](#)
- ◆ [ncurses_mouse_trafo](#)
- ◆ [ncurses_mouseinterval](#)
- ◆ [ncurses_mousemask](#)
- ◆ [ncurses_move](#)
- ◆ [ncurses_move_panel](#)
- ◆ [ncurses_mvaddch](#)
- ◆ [ncurses_mvaddchnstr](#)
- ◆ [ncurses_mvaddchstr](#)
- ◆ [ncurses_mvaddnstr](#)
- ◆ [ncurses_mvaddstr](#)
- ◆ [ncurses_mvcur](#)
- ◆ [ncurses_mvdclch](#)
- ◆ [ncurses_mvgetch](#)
- ◆ [ncurses_mvhline](#)
- ◆ [ncurses_mvinch](#)
- ◆ [ncurses_mvvline](#)
- ◆ [ncurses_mvwaddstr](#)

- ◆ [ncurses_napms](#)
- ◆ [ncurses_new_panel](#)
- ◆ [ncurses_newpad](#)
- ◆ [ncurses_newwin](#)
- ◆ [ncurses_nl](#)
- ◆ [ncurses_nocbreak](#)
- ◆ [ncurses_noecho](#)
- ◆ [ncurses_nonl](#)
- ◆ [ncurses_noqiflush](#)
- ◆ [ncurses_noraw](#)
- ◆ [ncurses_pair_content](#)
- ◆ [ncurses_panel_above](#)
- ◆ [ncurses_panel_below](#)
- ◆ [ncurses_panel_window](#)
- ◆ [ncurses_pnoutrefresh](#)
- ◆ [ncurses_prefresh](#)
- ◆ [ncurses_putp](#)
- ◆ [ncurses_qiflush](#)
- ◆ [ncurses_raw](#)
- ◆ [ncurses_refresh](#)
- ◆ [ncurses_replace_panel](#)
- ◆ [ncurses_reset_prog_mode](#)
- ◆ [ncurses_reset_shell_mode](#)
- ◆ [ncurses_resetty](#)
- ◆ [ncurses_savetty](#)
- ◆ [ncurses_scr_dump](#)
- ◆ [ncurses_scr_init](#)
- ◆ [ncurses_scr_restore](#)
- ◆ [ncurses_scr_set](#)
- ◆ [ncurses_scr1](#)
- ◆ [ncurses_show_panel](#)
- ◆ [ncurses_slk_attr](#)
- ◆ [ncurses_slk_attroff](#)
- ◆ [ncurses_slk_attron](#)
- ◆ [ncurses_slk_attrset](#)
- ◆ [ncurses_slk_clear](#)
- ◆ [ncurses_slk_color](#)
- ◆ [ncurses_slk_init](#)
- ◆ [ncurses_slk_noutrefresh](#)
- ◆ [ncurses_slk_refresh](#)
- ◆ [ncurses_slk_restore](#)
- ◆ [ncurses_slk_set](#)
- ◆ [ncurses_slk_touch](#)
- ◆ [ncurses_standend](#)
- ◆ [ncurses_standout](#)
- ◆ [ncurses_start_color](#)
- ◆ [ncurses_termattrs](#)
- ◆ [ncurses_termname](#)
- ◆ [ncurses_timeout](#)
- ◆ [ncurses_top_panel](#)
- ◆ [ncurses_typeahead](#)
- ◆ [ncurses_ungetch](#)
- ◆ [ncurses_ungetmouse](#)
- ◆ [ncurses_update_panels](#)

- ◆ [ncurses_use_default_colors](#)
- ◆ [ncurses_use_env](#)
- ◆ [ncurses_use_extended_names](#)
- ◆ [ncurses_vidattr](#)
- ◆ [ncurses_vline](#)
- ◆ [ncurses_waddch](#)
- ◆ [ncurses_waddstr](#)
- ◆ [ncurses_wattroff](#)
- ◆ [ncurses_wattron](#)
- ◆ [ncurses_wattrset](#)
- ◆ [ncurses_wborder](#)
- ◆ [ncurses_wclear](#)
- ◆ [ncurses_wcolor_set](#)
- ◆ [ncurses_werase](#)
- ◆ [ncurses_wgetch](#)
- ◆ [ncurses_whline](#)
- ◆ [ncurses_wmouse_trafo](#)
- ◆ [ncurses_wmove](#)
- ◆ [ncurses_wnoutrefresh](#)
- ◆ [ncurses_wrefresh](#)
- ◆ [ncurses_wstandend](#)
- ◆ [ncurses_wstandout](#)
- ◆ [ncurses_wvline](#)
- ◆ [next](#)
- ◆ [ngettext](#)
- ◆ [nl2br](#)
- ◆ [nl_langinfo](#)
- ◆ [notes_body](#)
- ◆ [notes_copy_db](#)
- ◆ [notes_create_db](#)
- ◆ [notes_create_note](#)
- ◆ [notes_drop_db](#)
- ◆ [notes_find_note](#)
- ◆ [notes_header_info](#)
- ◆ [notes_list_msgs](#)
- ◆ [notes_mark_read](#)
- ◆ [notes_mark_unread](#)
- ◆ [notes_nav_create](#)
- ◆ [notes_search](#)
- ◆ [notes_unread](#)
- ◆ [notes_version](#)
- ◆ [nsapi_request_headers](#)
- ◆ [nsapi_response_headers](#)
- ◆ [nsapi_virtual](#)
- ◆ [number_format](#)
- ○
- - ◆ [ob_clean](#)
 - ◆ [ob_end_clean](#)
 - ◆ [ob_end_flush](#)
 - ◆ [ob_flush](#)
 - ◆ [ob_get_clean](#)
 - ◆ [ob_get_contents](#)
 - ◆ [ob_get_flush](#)
 - ◆ [ob_get_length](#)

- ◆ [ob_get_level](#)
- ◆ [ob_get_status](#)
- ◆ [ob_gzhandler](#)
- ◆ [ob_iconv_handler](#)
- ◆ [ob_implicit_flush](#)
- ◆ [ob_list_handlers](#)
- ◆ [ob_start](#)
- ◆ [ob_tidyhandler](#)
- ◆ [oci-collection->append](#)
- ◆ [oci-collection->assign](#)
- ◆ [oci-collection->assignelem](#)
- ◆ [oci-collection->free](#)
- ◆ [oci-collection->getelem](#)
- ◆ [oci-collection->max](#)
- ◆ [oci-collection->size](#)
- ◆ [oci-collection->trim](#)
- ◆ [oci-lob->append](#)
- ◆ [oci-lob->close](#)
- ◆ [oci-lob->eof](#)
- ◆ [oci-lob->erase](#)
- ◆ [oci-lob->export](#)
- ◆ [oci-lob->flush](#)
- ◆ [oci-lob->free](#)
- ◆ [oci-lob->getbuffering](#)
- ◆ [oci-lob->import](#)
- ◆ [oci-lob->load](#)
- ◆ [oci-lob->read](#)
- ◆ [oci-lob->rewind](#)
- ◆ [oci-lob->save](#)
- ◆ [oci-lob->savefile](#)
- ◆ [oci-lob->seek](#)
- ◆ [oci-lob->setbuffering](#)
- ◆ [oci-lob->size](#)
- ◆ [oci-lob->tell](#)
- ◆ [oci-lob->truncate](#)
- ◆ [oci-lob->write](#)
- ◆ [oci-lob->writetemporary](#)
- ◆ [oci-lob->writetofile](#)
- ◆ [oci_bind_by_name](#)
- ◆ [oci_cancel](#)
- ◆ [oci_close](#)
- ◆ [oci_commit](#)
- ◆ [oci_connect](#)
- ◆ [oci_define_by_name](#)
- ◆ [oci_error](#)
- ◆ [oci_execute](#)
- ◆ [oci_fetch](#)
- ◆ [oci_fetch_all](#)
- ◆ [oci_fetch_array](#)
- ◆ [oci_fetch_assoc](#)
- ◆ [oci_fetch_object](#)
- ◆ [oci_fetch_row](#)
- ◆ [oci_field_is_null](#)
- ◆ [oci_field_name](#)

- ◆ [oci_field_precision](#)
- ◆ [oci_field_scale](#)
- ◆ [oci_field_size](#)
- ◆ [oci_field_type](#)
- ◆ [oci_field_type_raw](#)
- ◆ [oci_free_statement](#)
- ◆ [oci_internal_debug](#)
- ◆ [oci_lob_copy](#)
- ◆ [oci_lob_is_equal](#)
- ◆ [oci_new_collection](#)
- ◆ [oci_new_connect](#)
- ◆ [oci_new_cursor](#)
- ◆ [oci_new_descriptor](#)
- ◆ [oci_num_fields](#)
- ◆ [oci_num_rows](#)
- ◆ [oci_parse](#)
- ◆ [oci_password_change](#)
- ◆ [oci_pconnect](#)
- ◆ [oci_result](#)
- ◆ [oci_rollback](#)
- ◆ [oci_server_version](#)
- ◆ [oci_set_prefetch](#)
- ◆ [oci_statement_type](#)
- ◆ [ocibindbyname](#)
- ◆ [ocicancel](#)
- ◆ [ocicloselob](#)
- ◆ [ocicollappend](#)
- ◆ [ocicollassign](#)
- ◆ [ocicollassignelem](#)
- ◆ [ocicollgetelem](#)
- ◆ [ocicollmax](#)
- ◆ [ocicollsize](#)
- ◆ [ocicolltrim](#)
- ◆ [ocicolumnisnull](#)
- ◆ [ocicolumnname](#)
- ◆ [ocicolumnprecision](#)
- ◆ [ocicolumnscale](#)
- ◆ [ocicolumnsize](#)
- ◆ [ocicolumntype](#)
- ◆ [ocicolumntyperaw](#)
- ◆ [ocicommit](#)
- ◆ [ocidefinebyname](#)
- ◆ [ocierror](#)
- ◆ [ociexecute](#)
- ◆ [ocifetch](#)
- ◆ [ocifetchinto](#)
- ◆ [ocifetchstatement](#)
- ◆ [ocifreecollection](#)
- ◆ [ocifreecursor](#)
- ◆ [ocifreedesc](#)
- ◆ [ocifreestatement](#)
- ◆ [ociinternaldebug](#)
- ◆ [ociloadlob](#)
- ◆ [ocilogoff](#)

- ◆ [ocilogon](#)
- ◆ [ocinewcollection](#)
- ◆ [ocinewcursor](#)
- ◆ [ocinewscriptor](#)
- ◆ [ocinlogon](#)
- ◆ [ocinumcols](#)
- ◆ [ociparse](#)
- ◆ [ociplogon](#)
- ◆ [ociresult](#)
- ◆ [ocirollback](#)
- ◆ [ocirowcount](#)
- ◆ [ocisavelob](#)
- ◆ [ocisavelobfile](#)
- ◆ [ociserverversion](#)
- ◆ [ocisetprefetch](#)
- ◆ [ocistatementtype](#)
- ◆ [ociwritelobtofile](#)
- ◆ [ociwritetemporarylob](#)
- ◆ [octdec](#)
- ◆ [odbc_autocommit](#)
- ◆ [odbc_binmode](#)
- ◆ [odbc_close](#)
- ◆ [odbc_close_all](#)
- ◆ [odbc_columnprivileges](#)
- ◆ [odbc_columns](#)
- ◆ [odbc_commit](#)
- ◆ [odbc_connect](#)
- ◆ [odbc_cursor](#)
- ◆ [odbc_data_source](#)
- ◆ [odbc_do](#)
- ◆ [odbc_error](#)
- ◆ [odbc_errormsg](#)
- ◆ [odbc_exec](#)
- ◆ [odbc_execute](#)
- ◆ [odbc_fetch_array](#)
- ◆ [odbc_fetch_into](#)
- ◆ [odbc_fetch_object](#)
- ◆ [odbc_fetch_row](#)
- ◆ [odbc_field_len](#)
- ◆ [odbc_field_name](#)
- ◆ [odbc_field_num](#)
- ◆ [odbc_field_precision](#)
- ◆ [odbc_field_scale](#)
- ◆ [odbc_field_type](#)
- ◆ [odbc_foreignkeys](#)
- ◆ [odbc_free_result](#)
- ◆ [odbc_gettypeinfo](#)
- ◆ [odbc_longreadlen](#)
- ◆ [odbc_next_result](#)
- ◆ [odbc_num_fields](#)
- ◆ [odbc_num_rows](#)
- ◆ [odbc_pconnect](#)
- ◆ [odbc_prepare](#)
- ◆ [odbc_primarykeys](#)

- ◆ [odbc_procedurecolumns](#)
- ◆ [odbc_procedures](#)
- ◆ [odbc_result](#)
- ◆ [odbc_result_all](#)
- ◆ [odbc_rollback](#)
- ◆ [odbc_setoption](#)
- ◆ [odbc_specialcolumns](#)
- ◆ [odbc_statistics](#)
- ◆ [odbc_tableprivileges](#)
- ◆ [odbc_tables](#)
- ◆ [opental_buffer_create](#)
- ◆ [opental_buffer_data](#)
- ◆ [opental_buffer_destroy](#)
- ◆ [opental_buffer_get](#)
- ◆ [opental_buffer_loadwav](#)
- ◆ [opental_context_create](#)
- ◆ [opental_context_current](#)
- ◆ [opental_context_destroy](#)
- ◆ [opental_context_process](#)
- ◆ [opental_context_suspend](#)
- ◆ [opental_device_close](#)
- ◆ [opental_device_open](#)
- ◆ [opental_listener_get](#)
- ◆ [opental_listener_set](#)
- ◆ [opental_source_create](#)
- ◆ [opental_source_destroy](#)
- ◆ [opental_source_get](#)
- ◆ [opental_source_pause](#)
- ◆ [opental_source_play](#)
- ◆ [opental_source_rewind](#)
- ◆ [opental_source_set](#)
- ◆ [opental_source_stop](#)
- ◆ [opental_stream](#)
- ◆ [opendir](#)
- ◆ [openlog](#)
- ◆ [openssl_csr_export](#)
- ◆ [openssl_csr_export_to_file](#)
- ◆ [openssl_csr_new](#)
- ◆ [openssl_csr_sign](#)
- ◆ [openssl_error_string](#)
- ◆ [openssl_free_key](#)
- ◆ [openssl_get_privatekey](#)
- ◆ [openssl_get_publickey](#)
- ◆ [openssl_open](#)
- ◆ [openssl_pkcs7_decrypt](#)
- ◆ [openssl_pkcs7_encrypt](#)
- ◆ [openssl_pkcs7_sign](#)
- ◆ [openssl_pkcs7_verify](#)
- ◆ [openssl_pkey_export](#)
- ◆ [openssl_pkey_export_to_file](#)
- ◆ [openssl_pkey_free](#)
- ◆ [openssl_pkey_get_private](#)
- ◆ [openssl_pkey_get_public](#)
- ◆ [openssl_pkey_new](#)

- ◆ [openssl_private_decrypt](#)
- ◆ [openssl_private_encrypt](#)
- ◆ [openssl_public_decrypt](#)
- ◆ [openssl_public_encrypt](#)
- ◆ [openssl_seal](#)
- ◆ [openssl_sign](#)
- ◆ [openssl_verify](#)
- ◆ [openssl_x509_check_private_key](#)
- ◆ [openssl_x509_checkpurpose](#)
- ◆ [openssl_x509_export](#)
- ◆ [openssl_x509_export_to_file](#)
- ◆ [openssl_x509_free](#)
- ◆ [openssl_x509_parse](#)
- ◆ [openssl_x509_read](#)
- ◆ [ora_bind](#)
- ◆ [ora_close](#)
- ◆ [ora_columnname](#)
- ◆ [ora_columnsize](#)
- ◆ [ora_columntype](#)
- ◆ [ora_commit](#)
- ◆ [ora_commitoff](#)
- ◆ [ora_commiton](#)
- ◆ [ora_do](#)
- ◆ [ora_error](#)
- ◆ [ora_errorcode](#)
- ◆ [ora_exec](#)
- ◆ [ora_fetch](#)
- ◆ [ora_fetch_into](#)
- ◆ [ora_getcolumn](#)
- ◆ [ora_logoff](#)
- ◆ [ora_logon](#)
- ◆ [ora_numcols](#)
- ◆ [ora_numrows](#)
- ◆ [ora_open](#)
- ◆ [ora_parse](#)
- ◆ [ora_plogon](#)
- ◆ [ora_rollback](#)
- ◆ [ord](#)
- ◆ [output_add_rewrite_var](#)
- ◆ [output_reset_rewrite_vars](#)
- ◆ [overload](#)
- ◆ [override_function](#)
- ◆ [ovrimos_close](#)
- ◆ [ovrimos_commit](#)
- ◆ [ovrimos_connect](#)
- ◆ [ovrimos_cursor](#)
- ◆ [ovrimos_exec](#)
- ◆ [ovrimos_execute](#)
- ◆ [ovrimos_fetch_into](#)
- ◆ [ovrimos_fetch_row](#)
- ◆ [ovrimos_field_len](#)
- ◆ [ovrimos_field_name](#)
- ◆ [ovrimos_field_num](#)
- ◆ [ovrimos_field_type](#)

- ◆ [ovrimos_free_result](#)
- ◆ [ovrimos_longreadlen](#)
- ◆ [ovrimos_num_fields](#)
- ◆ [ovrimos_num_rows](#)
- ◆ [ovrimos_prepare](#)
- ◆ [ovrimos_result](#)
- ◆ [ovrimos_result_all](#)
- ◆ [ovrimos_rollback](#)
- **P**
- - ◆ [pack](#)
 - ◆ [parentiterator::getchildren](#)
 - ◆ [parentiterator::haschildren](#)
 - ◆ [parentiterator::next](#)
 - ◆ [parentiterator::rewind](#)
 - ◆ [parse_ini_file](#)
 - ◆ [parse_str](#)
 - ◆ [parse_url](#)
 - ◆ [parsekit_compile_file](#)
 - ◆ [parsekit_compile_string](#)
 - ◆ [parsekit_func_arginfo](#)
 - ◆ [passthru](#)
 - ◆ [pathinfo](#)
 - ◆ [pclose](#)
 - ◆ [pcntl_alarm](#)
 - ◆ [pcntl_exec](#)
 - ◆ [pcntl_fork](#)
 - ◆ [pcntl_getpriority](#)
 - ◆ [pcntl_setpriority](#)
 - ◆ [pcntl_signal](#)
 - ◆ [pcntl_wait](#)
 - ◆ [pcntl_waitpid](#)
 - ◆ [pcntl_wexitstatus](#)
 - ◆ [pcntl_wifexited](#)
 - ◆ [pcntl_wifsignaled](#)
 - ◆ [pcntl_wifstopped](#)
 - ◆ [pcntl_wstopsig](#)
 - ◆ [pcntl_wtermsig](#)
 - ◆ [pdf_add_annotation](#)
 - ◆ [pdf_add_bookmark](#)
 - ◆ [pdf_add_launchlink](#)
 - ◆ [pdf_add_loclink](#)
 - ◆ [pdf_add_note](#)
 - ◆ [pdf_add_outline](#)
 - ◆ [pdf_add_pdflink](#)
 - ◆ [pdf_add_thumbnail](#)
 - ◆ [pdf_add_weblink](#)
 - ◆ [pdf_arc](#)
 - ◆ [pdf_arcn](#)
 - ◆ [pdf_attach_file](#)
 - ◆ [pdf_begin_page](#)
 - ◆ [pdf_begin_pattern](#)
 - ◆ [pdf_begin_template](#)
 - ◆ [pdf_circle](#)
 - ◆ [pdf_clip](#)

- ◆ [pdf_close](#)
- ◆ [pdf_close_image](#)
- ◆ [pdf_close_pdi](#)
- ◆ [pdf_close_pdi_page](#)
- ◆ [pdf_closepath](#)
- ◆ [pdf_closepath_fill_stroke](#)
- ◆ [pdf_closepath_stroke](#)
- ◆ [pdf_concat](#)
- ◆ [pdf_continue_text](#)
- ◆ [pdf_curveto](#)
- ◆ [pdf_delete](#)
- ◆ [pdf_end_page](#)
- ◆ [pdf_end_pattern](#)
- ◆ [pdf_end_template](#)
- ◆ [pdf_endpath](#)
- ◆ [pdf_fill](#)
- ◆ [pdf_fill_stroke](#)
- ◆ [pdf_findfont](#)
- ◆ [pdf_get_buffer](#)
- ◆ [pdf_get_font](#)
- ◆ [pdf_get_fontname](#)
- ◆ [pdf_get_fontsize](#)
- ◆ [pdf_get_image_height](#)
- ◆ [pdf_get_image_width](#)
- ◆ [pdf_get_majorversion](#)
- ◆ [pdf_get_minorversion](#)
- ◆ [pdf_get_parameter](#)
- ◆ [pdf_get_pdi_parameter](#)
- ◆ [pdf_get_pdi_value](#)
- ◆ [pdf_get_value](#)
- ◆ [pdf_initgraphics](#)
- ◆ [pdf_lineto](#)
- ◆ [pdf_makespotcolor](#)
- ◆ [pdf_moveto](#)
- ◆ [pdf_new](#)
- ◆ [pdf_open](#)
- ◆ [pdf_open_ccitt](#)
- ◆ [pdf_open_file](#)
- ◆ [pdf_open_gif](#)
- ◆ [pdf_open_image](#)
- ◆ [pdf_open_image_file](#)
- ◆ [pdf_open_jpeg](#)
- ◆ [pdf_open_memory_image](#)
- ◆ [pdf_open_pdi](#)
- ◆ [pdf_open_pdi_page](#)
- ◆ [pdf_open_png](#)
- ◆ [pdf_open_tiff](#)
- ◆ [pdf_place_image](#)
- ◆ [pdf_place_pdi_page](#)
- ◆ [pdf_rect](#)
- ◆ [pdf_restore](#)
- ◆ [pdf_rotate](#)
- ◆ [pdf_save](#)
- ◆ [pdf_scale](#)

- ◆ [pdf_set_border_color](#)
- ◆ [pdf_set_border_dash](#)
- ◆ [pdf_set_border_style](#)
- ◆ [pdf_set_char_spacing](#)
- ◆ [pdf_set_duration](#)
- ◆ [pdf_set_font](#)
- ◆ [pdf_set_horiz_scaling](#)
- ◆ [pdf_set_info](#)
- ◆ [pdf_set_info_author](#)
- ◆ [pdf_set_info_creator](#)
- ◆ [pdf_set_info_keywords](#)
- ◆ [pdf_set_info_subject](#)
- ◆ [pdf_set_info_title](#)
- ◆ [pdf_set_leading](#)
- ◆ [pdf_set_parameter](#)
- ◆ [pdf_set_text_matrix](#)
- ◆ [pdf_set_text_pos](#)
- ◆ [pdf_set_text_rendering](#)
- ◆ [pdf_set_text_rise](#)
- ◆ [pdf_set_value](#)
- ◆ [pdf_set_word_spacing](#)
- ◆ [pdf_setcolor](#)
- ◆ [pdf_setdash](#)
- ◆ [pdf_setflat](#)
- ◆ [pdf_setfont](#)
- ◆ [pdf_setgray](#)
- ◆ [pdf_setgray_fill](#)
- ◆ [pdf_setgray_stroke](#)
- ◆ [pdf_setlinecap](#)
- ◆ [pdf_setlinejoin](#)
- ◆ [pdf_setlinewidth](#)
- ◆ [pdf_setmatrix](#)
- ◆ [pdf_setmiterlimit](#)
- ◆ [pdf_setpolydash](#)
- ◆ [pdf_setrgbcolor](#)
- ◆ [pdf_setrgbcolor_fill](#)
- ◆ [pdf_setrgbcolor_stroke](#)
- ◆ [pdf_show](#)
- ◆ [pdf_show_boxed](#)
- ◆ [pdf_show_xy](#)
- ◆ [pdf_skew](#)
- ◆ [pdf_stringwidth](#)
- ◆ [pdf_stroke](#)
- ◆ [pdf_translate](#)
- ◆ [pdo::__construct](#)
- ◆ [pdo::beginTransaction](#)
- ◆ [pdo::commit](#)
- ◆ [pdo::errorCode](#)
- ◆ [pdo::errorInfo](#)
- ◆ [pdo::exec](#)
- ◆ [pdo::getAttribute](#)
- ◆ [pdo::getAvailableDrivers](#)
- ◆ [pdo::lastInsertId](#)
- ◆ [pdo::prepare](#)

- ◆ [pdo::query](#)
- ◆ [pdo::quote](#)
- ◆ [pdo::rollback](#)
- ◆ [pdo::setAttribute](#)
- ◆ [pdostatement::bindcolumn](#)
- ◆ [pdostatement::bindParam](#)
- ◆ [pdostatement::bindValue](#)
- ◆ [pdostatement::closecursor](#)
- ◆ [pdostatement::columncount](#)
- ◆ [pdostatement::errorcode](#)
- ◆ [pdostatement::errorinfo](#)
- ◆ [pdostatement::execute](#)
- ◆ [pdostatement::fetch](#)
- ◆ [pdostatement::fetchall](#)
- ◆ [pdostatement::fetchcolumn](#)
- ◆ [pdostatement::getAttribute](#)
- ◆ [pdostatement::getColumnmeta](#)
- ◆ [pdostatement::nextrowset](#)
- ◆ [pdostatement::rowcount](#)
- ◆ [pdostatement::setAttribute](#)
- ◆ [pdostatement::setfetchmode](#)
- ◆ [pfpro_cleanup](#)
- ◆ [pfpro_init](#)
- ◆ [pfpro_process](#)
- ◆ [pfpro_process_raw](#)
- ◆ [pfpro_version](#)
- ◆ [pfsockopen](#)
- ◆ [pg_affected_rows](#)
- ◆ [pg_cancel_query](#)
- ◆ [pg_client_encoding](#)
- ◆ [pg_close](#)
- ◆ [pg_connect](#)
- ◆ [pg_connection_busy](#)
- ◆ [pg_connection_reset](#)
- ◆ [pg_connection_status](#)
- ◆ [pg_convert](#)
- ◆ [pg_copy_from](#)
- ◆ [pg_copy_to](#)
- ◆ [pg_dbname](#)
- ◆ [pg_delete](#)
- ◆ [pg_end_copy](#)
- ◆ [pg_escape_bytea](#)
- ◆ [pg_escape_string](#)
- ◆ [pg_execute](#)
- ◆ [pg_fetch_all](#)
- ◆ [pg_fetch_all_columns](#)
- ◆ [pg_fetch_array](#)
- ◆ [pg_fetch_assoc](#)
- ◆ [pg_fetch_object](#)
- ◆ [pg_fetch_result](#)
- ◆ [pg_fetch_row](#)
- ◆ [pg_field_is_null](#)
- ◆ [pg_field_name](#)
- ◆ [pg_field_num](#)

- ◆ [pg_field_prtlen](#)
- ◆ [pg_field_size](#)
- ◆ [pg_field_type](#)
- ◆ [pg_field_type_oid](#)
- ◆ [pg_free_result](#)
- ◆ [pg_get_notify](#)
- ◆ [pg_get_pid](#)
- ◆ [pg_get_result](#)
- ◆ [pg_host](#)
- ◆ [pg_insert](#)
- ◆ [pg_last_error](#)
- ◆ [pg_last_notice](#)
- ◆ [pg_last_oid](#)
- ◆ [pg_lo_close](#)
- ◆ [pg_lo_create](#)
- ◆ [pg_lo_export](#)
- ◆ [pg_lo_import](#)
- ◆ [pg_lo_open](#)
- ◆ [pg_lo_read](#)
- ◆ [pg_lo_read_all](#)
- ◆ [pg_lo_seek](#)
- ◆ [pg_lo_tell](#)
- ◆ [pg_lo_unlink](#)
- ◆ [pg_lo_write](#)
- ◆ [pg_meta_data](#)
- ◆ [pg_num_fields](#)
- ◆ [pg_num_rows](#)
- ◆ [pg_options](#)
- ◆ [pg_parameter_status](#)
- ◆ [pg_pconnect](#)
- ◆ [pg_ping](#)
- ◆ [pg_port](#)
- ◆ [pg_prepare](#)
- ◆ [pg_put_line](#)
- ◆ [pg_query](#)
- ◆ [pg_query_params](#)
- ◆ [pg_result_error](#)
- ◆ [pg_result_error_field](#)
- ◆ [pg_result_seek](#)
- ◆ [pg_result_status](#)
- ◆ [pg_select](#)
- ◆ [pg_send_execute](#)
- ◆ [pg_send_prepare](#)
- ◆ [pg_send_query](#)
- ◆ [pg_send_query_params](#)
- ◆ [pg_set_client_encoding](#)
- ◆ [pg_set_error_verbosity](#)
- ◆ [pg_trace](#)
- ◆ [pg_transaction_status](#)
- ◆ [pg_tty](#)
- ◆ [pg_unescape_bytea](#)
- ◆ [pg_untrace](#)
- ◆ [pg_update](#)
- ◆ [pg_version](#)

- ◆ [php_check_syntax](#)
- ◆ [php_ini_scanned_files](#)
- ◆ [php_logo_guid](#)
- ◆ [php_sapi_name](#)
- ◆ [php_strip_whitespace](#)
- ◆ [php_uname](#)
- ◆ [phpcredits](#)
- ◆ [phpinfo](#)
- ◆ [phpversion](#)
- ◆ [pi](#)
- ◆ [png2wbmp](#)
- ◆ [popen](#)
- ◆ [pos](#)
- ◆ [posix_access](#)
- ◆ [posix_ctermid](#)
- ◆ [posix_get_last_error](#)
- ◆ [posix_getcwd](#)
- ◆ [posix_getegid](#)
- ◆ [posix_geteuid](#)
- ◆ [posix_getgid](#)
- ◆ [posix_getrgid](#)
- ◆ [posix_getrnam](#)
- ◆ [posix_getgroups](#)
- ◆ [posix_getlogin](#)
- ◆ [posix_getpgid](#)
- ◆ [posix_getpgrp](#)
- ◆ [posix_getpid](#)
- ◆ [posix_getppid](#)
- ◆ [posix_getpwnam](#)
- ◆ [posix_getpwuid](#)
- ◆ [posix_getrlimit](#)
- ◆ [posix_getsid](#)
- ◆ [posix_getuid](#)
- ◆ [posix_isatty](#)
- ◆ [posix_kill](#)
- ◆ [posix_mkfifo](#)
- ◆ [posix_mknod](#)
- ◆ [posix_setegid](#)
- ◆ [posix seteuid](#)
- ◆ [posix_setgid](#)
- ◆ [posix_setpgid](#)
- ◆ [posix_setsid](#)
- ◆ [posix_setuid](#)
- ◆ [posix_strerror](#)
- ◆ [posix_times](#)
- ◆ [posix_ttyname](#)
- ◆ [posix_uname](#)
- ◆ [pow](#)
- ◆ [preg_grep](#)
- ◆ [preg_match](#)
- ◆ [preg_match_all](#)
- ◆ [preg_quote](#)
- ◆ [preg_replace](#)
- ◆ [preg_replace_callback](#)

- ◆ [preg_split](#)
- ◆ [prev](#)
- ◆ [print](#)
- ◆ [print_r](#)
- ◆ [printer_abort](#)
- ◆ [printer_close](#)
- ◆ [printer_create_brush](#)
- ◆ [printer_create_dc](#)
- ◆ [printer_create_font](#)
- ◆ [printer_create_pen](#)
- ◆ [printer_delete_brush](#)
- ◆ [printer_delete_dc](#)
- ◆ [printer_delete_font](#)
- ◆ [printer_delete_pen](#)
- ◆ [printer_draw_bmp](#)
- ◆ [printer_draw_chord](#)
- ◆ [printer_draw_ellipse](#)
- ◆ [printer_draw_line](#)
- ◆ [printer_draw_pie](#)
- ◆ [printer_draw_rectangle](#)
- ◆ [printer_draw_roundrect](#)
- ◆ [printer_draw_text](#)
- ◆ [printer_end_doc](#)
- ◆ [printer_end_page](#)
- ◆ [printer_get_option](#)
- ◆ [printer_list](#)
- ◆ [printer_logical_fontheight](#)
- ◆ [printer_open](#)
- ◆ [printer_select_brush](#)
- ◆ [printer_select_font](#)
- ◆ [printer_select_pen](#)
- ◆ [printer_set_option](#)
- ◆ [printer_start_doc](#)
- ◆ [printer_start_page](#)
- ◆ [printer_write](#)
- ◆ [printf](#)
- ◆ [proc_close](#)
- ◆ [proc_get_status](#)
- ◆ [proc_nice](#)
- ◆ [proc_open](#)
- ◆ [proc_terminate](#)
- ◆ [property_exists](#)
- ◆ [ps_add_bookmark](#)
- ◆ [ps_add_launchlink](#)
- ◆ [ps_add_locallink](#)
- ◆ [ps_add_note](#)
- ◆ [ps_add_pdflink](#)
- ◆ [ps_add_weblink](#)
- ◆ [ps_arc](#)
- ◆ [ps_arcn](#)
- ◆ [ps_begin_page](#)
- ◆ [ps_begin_pattern](#)
- ◆ [ps_begin_template](#)
- ◆ [ps_circle](#)

- ◆ [ps_clip](#)
- ◆ [ps_close](#)
- ◆ [ps_close_image](#)
- ◆ [ps_closepath](#)
- ◆ [ps_closepath_stroke](#)
- ◆ [ps_continue_text](#)
- ◆ [ps_curveto](#)
- ◆ [ps_delete](#)
- ◆ [ps_end_page](#)
- ◆ [ps_end_pattern](#)
- ◆ [ps_end_template](#)
- ◆ [ps_fill](#)
- ◆ [ps_fill_stroke](#)
- ◆ [ps_findfont](#)
- ◆ [ps_get_buffer](#)
- ◆ [ps_get_parameter](#)
- ◆ [ps_get_value](#)
- ◆ [ps_hyphenate](#)
- ◆ [ps_lineto](#)
- ◆ [ps_makespotcolor](#)
- ◆ [ps_moveto](#)
- ◆ [ps_new](#)
- ◆ [ps_open_file](#)
- ◆ [ps_open_image](#)
- ◆ [ps_open_image_file](#)
- ◆ [ps_place_image](#)
- ◆ [ps_rect](#)
- ◆ [ps_restore](#)
- ◆ [ps_rotate](#)
- ◆ [ps_save](#)
- ◆ [ps_scale](#)
- ◆ [ps_set_border_color](#)
- ◆ [ps_set_border_dash](#)
- ◆ [ps_set_border_style](#)
- ◆ [ps_set_info](#)
- ◆ [ps_set_parameter](#)
- ◆ [ps_set_text_pos](#)
- ◆ [ps_set_value](#)
- ◆ [ps_setcolor](#)
- ◆ [ps_setdash](#)
- ◆ [ps_setflat](#)
- ◆ [ps_setfont](#)
- ◆ [ps_setgray](#)
- ◆ [ps_setlinecap](#)
- ◆ [ps_setlinejoin](#)
- ◆ [ps_setlinewidth](#)
- ◆ [ps_setmiterlimit](#)
- ◆ [ps_setpolydash](#)
- ◆ [ps_shading](#)
- ◆ [ps_shading_pattern](#)
- ◆ [ps_shfill](#)
- ◆ [ps_show](#)
- ◆ [ps_show_boxed](#)
- ◆ [ps_show_xy](#)

- ◆ [ps_string_geometry](#)
- ◆ [ps_stringwidth](#)
- ◆ [ps_stroke](#)
- ◆ [ps_symbol](#)
- ◆ [ps_symbol_name](#)
- ◆ [ps_symbol_width](#)
- ◆ [ps_translate](#)
- ◆ [pspell_add_to_personal](#)
- ◆ [pspell_add_to_session](#)
- ◆ [pspell_check](#)
- ◆ [pspell_clear_session](#)
- ◆ [pspell_config_create](#)
- ◆ [pspell_config_data_dir](#)
- ◆ [pspell_config_dict_dir](#)
- ◆ [pspell_config_ignore](#)
- ◆ [pspell_config_mode](#)
- ◆ [pspell_config_personal](#)
- ◆ [pspell_config_repl](#)
- ◆ [pspell_config_runtogether](#)
- ◆ [pspell_config_save_repl](#)
- ◆ [pspell_new](#)
- ◆ [pspell_new_config](#)
- ◆ [pspell_new_personal](#)
- ◆ [pspell_save_wordlist](#)
- ◆ [pspell_store_replacement](#)
- ◆ [pspell_suggest](#)
- ◆ [putenv](#)
- ◆ [px_close](#)
- ◆ [px_create_fp](#)
- ◆ [px_delete](#)
- ◆ [px_get_field](#)
- ◆ [px_get_info](#)
- ◆ [px_get_parameter](#)
- ◆ [px_get_record](#)
- ◆ [px_get_schema](#)
- ◆ [px_get_value](#)
- ◆ [px_new](#)
- ◆ [px_numfields](#)
- ◆ [px_numrecords](#)
- ◆ [px_open_fp](#)
- ◆ [px_put_record](#)
- ◆ [px_set_blob_file](#)
- ◆ [px_set_parameter](#)
- ◆ [px_set_tablename](#)
- ◆ [px_set_targetencoding](#)
- ◆ [px_set_value](#)
- ◆ [px_timestamp2string](#)
- **Q**
- ◆ [qdom_error](#)
- ◆ [qdom_tree](#)
- ◆ [quoted_printable_decode](#)
- ◆ [quotemeta](#)
- **R**
- ◆ [rad2deg](#)

- ◆ [radius_acct_open](#)
- ◆ [radius_add_server](#)
- ◆ [radius_auth_open](#)
- ◆ [radius_close](#)
- ◆ [radius_config](#)
- ◆ [radius_create_request](#)
- ◆ [radius_cvt_addr](#)
- ◆ [radius_cvt_int](#)
- ◆ [radius_cvt_string](#)
- ◆ [radius_demangle](#)
- ◆ [radius_demangle_mppe_key](#)
- ◆ [radius_get_attr](#)
- ◆ [radius_get_vendor_attr](#)
- ◆ [radius_put_addr](#)
- ◆ [radius_put_attr](#)
- ◆ [radius_put_int](#)
- ◆ [radius_put_string](#)
- ◆ [radius_put_vendor_addr](#)
- ◆ [radius_put_vendor_attr](#)
- ◆ [radius_put_vendor_int](#)
- ◆ [radius_put_vendor_string](#)
- ◆ [radius_request_authenticator](#)
- ◆ [radius_send_request](#)
- ◆ [radius_server_secret](#)
- ◆ [radius_strerror](#)
- ◆ [rand](#)
- ◆ [range](#)
- ◆ [rar::extract](#)
- ◆ [rar::getattr](#)
- ◆ [rar::getcrc](#)
- ◆ [rar::getfiletime](#)
- ◆ [rar::gethostos](#)
- ◆ [rar::getmethod](#)
- ◆ [rar::getname](#)
- ◆ [rar::getpackedsize](#)
- ◆ [rar::getunpackedsize](#)
- ◆ [rar::getversion](#)
- ◆ [rar_close](#)
- ◆ [rar_entry_get](#)
- ◆ [rar_list](#)
- ◆ [rar_open](#)
- ◆ [rawurldecode](#)
- ◆ [rawurlencode](#)
- ◆ [read_exif_data](#)
- ◆ [readdir](#)
- ◆ [readfile](#)
- ◆ [readgzfile](#)
- ◆ [readline](#)
- ◆ [readline_add_history](#)
- ◆ [readline_callback_handler_install](#)
- ◆ [readline_callback_handler_remove](#)
- ◆ [readline_callback_read_char](#)
- ◆ [readline_clear_history](#)
- ◆ [readline_completion_function](#)

- ◆ [readline_info](#)
- ◆ [readline_list_history](#)
- ◆ [readline_on_new_line](#)
- ◆ [readline_read_history](#)
- ◆ [readline_redisplay](#)
- ◆ [readline_write_history](#)
- ◆ [readlink](#)
- ◆ [realpath](#)
- ◆ [recode](#)
- ◆ [recode_file](#)
- ◆ [recode_string](#)
- ◆ [recursivedirectoryiterator::getchildren](#)
- ◆ [recursivedirectoryiterator::haschildren](#)
- ◆ [recursivedirectoryiterator::key](#)
- ◆ [recursivedirectoryiterator::next](#)
- ◆ [recursivedirectoryiterator::rewind](#)
- ◆ [recursiveiteratoriterator::current](#)
- ◆ [recursiveiteratoriterator::getdepth](#)
- ◆ [recursiveiteratoriterator::getsubiterator](#)
- ◆ [recursiveiteratoriterator::key](#)
- ◆ [recursiveiteratoriterator::next](#)
- ◆ [recursiveiteratoriterator::rewind](#)
- ◆ [recursiveiteratoriterator::valid](#)
- ◆ [register_shutdown_function](#)
- ◆ [register_tick_function](#)
- ◆ [rename](#)
- ◆ [rename_function](#)
- ◆ [reset](#)
- ◆ [restore_error_handler](#)
- ◆ [restore_exception_handler](#)
- ◆ [restore_include_path](#)
- ◆ [result->current_field](#)
- ◆ [result->current_field](#)
- ◆ [result->data_seek](#)
- ◆ [result->data_seek](#)
- ◆ [result->fetch_array](#)
- ◆ [result->fetch_array](#)
- ◆ [result->fetch_field](#)
- ◆ [result->fetch_field](#)
- ◆ [result->fetch_field_direct](#)
- ◆ [result->fetch_field_direct](#)
- ◆ [result->fetch_fields](#)
- ◆ [result->fetch_fields](#)
- ◆ [result->fetch_object](#)
- ◆ [result->fetch_object](#)
- ◆ [result->fetch_row](#)
- ◆ [result->fetch_row](#)
- ◆ [result->field_count](#)
- ◆ [result->field_count](#)
- ◆ [result->field_seek](#)
- ◆ [result->field_seek](#)
- ◆ [result->free](#)
- ◆ [result->free](#)
- ◆ [result->lengths](#)

- ◆ [result->lengths](#)
 - ◆ [rewind](#)
 - ◆ [rewinddir](#)
 - ◆ [rmdir](#)
 - ◆ [round](#)
 - ◆ [rsort](#)
 - ◆ [rtrim](#)
 - ◆ [runkit_class_adopt](#)
 - ◆ [runkit_class_emancipate](#)
 - ◆ [runkit_constant_add](#)
 - ◆ [runkit_constant_redefine](#)
 - ◆ [runkit_constant_remove](#)
 - ◆ [runkit_function_add](#)
 - ◆ [runkit_function_copy](#)
 - ◆ [runkit_function_redefine](#)
 - ◆ [runkit_function_remove](#)
 - ◆ [runkit_function_rename](#)
 - ◆ [runkit_import](#)
 - ◆ [runkit_lint](#)
 - ◆ [runkit_lint_file](#)
 - ◆ [runkit_method_add](#)
 - ◆ [runkit_method_copy](#)
 - ◆ [runkit_method_redefine](#)
 - ◆ [runkit_method_remove](#)
 - ◆ [runkit_method_rename](#)
 - ◆ [runkit_return_value_used](#)
 - ◆ [runkit_sandbox_output_handler](#)
 - ◆ [runkit_superglobals](#)
- S
- ◆ [scandir](#)
 - ◆ [sem_acquire](#)
 - ◆ [sem_get](#)
 - ◆ [sem_release](#)
 - ◆ [sem_remove](#)
 - ◆ [serialize](#)
 - ◆ [sesam_affected_rows](#)
 - ◆ [sesam_commit](#)
 - ◆ [sesam_connect](#)
 - ◆ [sesam_diagnostic](#)
 - ◆ [sesam_disconnect](#)
 - ◆ [sesam_errormsg](#)
 - ◆ [sesam_execimm](#)
 - ◆ [sesam_fetch_array](#)
 - ◆ [sesam_fetch_result](#)
 - ◆ [sesam_fetch_row](#)
 - ◆ [sesam_field_array](#)
 - ◆ [sesam_field_name](#)
 - ◆ [sesam_free_result](#)
 - ◆ [sesam_num_fields](#)
 - ◆ [sesam_query](#)
 - ◆ [sesam_rollback](#)
 - ◆ [sesam_seek_row](#)
 - ◆ [sesam_settransaction](#)
 - ◆ [session_cache_expire](#)

- ◆ [session_cache_limiter](#)
- ◆ [session_commit](#)
- ◆ [session_decode](#)
- ◆ [session_destroy](#)
- ◆ [session_encode](#)
- ◆ [session_get_cookie_params](#)
- ◆ [session_id](#)
- ◆ [session_is_registered](#)
- ◆ [session_module_name](#)
- ◆ [session_name](#)
- ◆ [session_regenerate_id](#)
- ◆ [session_register](#)
- ◆ [session_save_path](#)
- ◆ [session_set_cookie_params](#)
- ◆ [session_set_save_handler](#)
- ◆ [session_start](#)
- ◆ [session_unregister](#)
- ◆ [session_unset](#)
- ◆ [session_write_close](#)
- ◆ [set_error_handler](#)
- ◆ [set_exception_handler](#)
- ◆ [set_file_buffer](#)
- ◆ [set_include_path](#)
- ◆ [set_magic_quotes_runtime](#)
- ◆ [set_time_limit](#)
- ◆ [setcookie](#)
- ◆ [setlocale](#)
- ◆ [setrawcookie](#)
- ◆ [settype](#)
- ◆ [sha1](#)
- ◆ [sha1_file](#)
- ◆ [shell_exec](#)
- ◆ [shm_attach](#)
- ◆ [shm_detach](#)
- ◆ [shm_get_var](#)
- ◆ [shm_put_var](#)
- ◆ [shm_remove](#)
- ◆ [shm_remove_var](#)
- ◆ [shmop_close](#)
- ◆ [shmop_delete](#)
- ◆ [shmop_open](#)
- ◆ [shmop_read](#)
- ◆ [shmop_size](#)
- ◆ [shmop_write](#)
- ◆ [show_source](#)
- ◆ [shuffle](#)
- ◆ [similar_text](#)
- ◆ [simplexml_import_dom](#)
- ◆ [simplexml_load_file](#)
- ◆ [simplexml_load_string](#)
- ◆ [simplexmlelement->asxml](#)
- ◆ [simplexmlelement->attributes](#)
- ◆ [simplexmlelement->children](#)
- ◆ [simplexmlelement->xpath](#)

- ◆ [simplexmliterator::current](#)
- ◆ [simplexmliterator::getchildren](#)
- ◆ [simplexmliterator::haschildren](#)
- ◆ [simplexmliterator::key](#)
- ◆ [simplexmliterator::next](#)
- ◆ [simplexmliterator::rewind](#)
- ◆ [simplexmliterator::valid](#)
- ◆ [sin](#)
- ◆ [sinh](#)
- ◆ [sizeof](#)
- ◆ [sleep](#)
- ◆ [snmp_get_quick_print](#)
- ◆ [snmp_get_valueretrieval](#)
- ◆ [snmp_read_mib](#)
- ◆ [snmp_set_enum_print](#)
- ◆ [snmp_set_oid_numeric_print](#)
- ◆ [snmp_set_quick_print](#)
- ◆ [snmp_set_valueretrieval](#)
- ◆ [snmpget](#)
- ◆ [snmpgetnext](#)
- ◆ [snmprealwalk](#)
- ◆ [snmpset](#)
- ◆ [snmpwalk](#)
- ◆ [snmpwalkoid](#)
- ◆ [soapclient->__call\(\)](#)
- ◆ [soapclient->__construct\(\)](#)
- ◆ [soapclient->__dorequest\(\)](#)
- ◆ [soapclient->__getfunctions\(\)](#)
- ◆ [soapclient->__getlastrequest\(\)](#)
- ◆ [soapclient->__getlastrequestheaders\(\)](#)
- ◆ [soapclient->__getlastresponse\(\)](#)
- ◆ [soapclient->__getlastresponseheaders\(\)](#)
- ◆ [soapclient->__gettypes\(\)](#)
- ◆ [soapclient->__setcookie\(\)](#)
- ◆ [soapclient->__soapcall\(\)](#)
- ◆ [soapfault->__construct\(\)](#)
- ◆ [soapheader->__construct\(\)](#)
- ◆ [soapparam->__construct\(\)](#)
- ◆ [soapserver->__construct\(\)](#)
- ◆ [soapserver->addfunction\(\)](#)
- ◆ [soapserver->fault\(\)](#)
- ◆ [soapserver->getfunctions\(\)](#)
- ◆ [soapserver->handle\(\)](#)
- ◆ [soapserver->setclass\(\)](#)
- ◆ [soapserver->setpersistence\(\)](#)
- ◆ [soapvar->__construct\(\)](#)
- ◆ [socket_accept](#)
- ◆ [socket_bind](#)
- ◆ [socket_clear_error](#)
- ◆ [socket_close](#)
- ◆ [socket_connect](#)
- ◆ [socket_create](#)
- ◆ [socket_create_listen](#)
- ◆ [socket_create_pair](#)

- ◆ [socket_get_option](#)
- ◆ [socket_get_status](#)
- ◆ [socket_getpeername](#)
- ◆ [socket_getsockname](#)
- ◆ [socket_last_error](#)
- ◆ [socket_listen](#)
- ◆ [socket_read](#)
- ◆ [socket_recv](#)
- ◆ [socket_recvfrom](#)
- ◆ [socket_select](#)
- ◆ [socket_send](#)
- ◆ [socket_sendto](#)
- ◆ [socket_set_block](#)
- ◆ [socket_set_blocking](#)
- ◆ [socket_set_nonblock](#)
- ◆ [socket_set_option](#)
- ◆ [socket_set_timeout](#)
- ◆ [socket_shutdown](#)
- ◆ [socket_strerror](#)
- ◆ [socket_write](#)
- ◆ [sort](#)
- ◆ [soundex](#)
- ◆ [spl_classes](#)
- ◆ [split](#)
- ◆ [spliti](#)
- ◆ [sprintf](#)
- ◆ [sql_regcase](#)
- ◆ [sqlite_close](#)
- ◆ [sqlite_error_string](#)
- ◆ [sqlite_escape_string](#)
- ◆ [sqlite_factory](#)
- ◆ [sqlite_fetch_string](#)
- ◆ [sqlite_has_more](#)
- ◆ [sqlite_libencoding](#)
- ◆ [sqlite_libversion](#)
- ◆ [sqlite_open](#)
- ◆ [sqlite_popen](#)
- ◆ [sqlite_udf_decode_binary](#)
- ◆ [sqlite_udf_encode_binary](#)
- ◆ [sqlitedatabase->arrayquery](#)
- ◆ [sqlitedatabase->busytimeout](#)
- ◆ [sqlitedatabase->changes](#)
- ◆ [sqlitedatabase->createaggregate](#)
- ◆ [sqlitedatabase->createfunction](#)
- ◆ [sqlitedatabase->exec](#)
- ◆ [sqlitedatabase->fetchcolumnntypes](#)
- ◆ [sqlitedatabase->lasterror](#)
- ◆ [sqlitedatabase->lastinsertrowid](#)
- ◆ [sqlitedatabase->query](#)
- ◆ [sqlitedatabase->singlequery](#)
- ◆ [sqlitedatabase->unbufferedquery](#)
- ◆ [sqliteresult->hasprev](#)
- ◆ [sqliteresult->key](#)
- ◆ [sqliteresult->numrows](#)

- ◆ [sqliteresult->prev](#)
- ◆ [sqliteresult->rewind](#)
- ◆ [sqliteresult->seek](#)
- ◆ [sqliteunbuffered->column](#)
- ◆ [sqliteunbuffered->current](#)
- ◆ [sqliteunbuffered->fetch](#)
- ◆ [sqliteunbuffered->fetchall](#)
- ◆ [sqliteunbuffered->fetchobject](#)
- ◆ [sqliteunbuffered->fetchsingle](#)
- ◆ [sqliteunbuffered->fieldname](#)
- ◆ [sqliteunbuffered->next](#)
- ◆ [sqliteunbuffered->numfields](#)
- ◆ [sqliteunbuffered->valid](#)
- ◆ [sqrt](#)
- ◆ [srand](#)
- ◆ [sscanf](#)
- ◆ [ssh2_auth_hostbased_file](#)
- ◆ [ssh2_auth_none](#)
- ◆ [ssh2_auth_password](#)
- ◆ [ssh2_auth_pubkey_file](#)
- ◆ [ssh2_connect](#)
- ◆ [ssh2_exec](#)
- ◆ [ssh2_fetch_stream](#)
- ◆ [ssh2_fingerprint](#)
- ◆ [ssh2_methods_negotiated](#)
- ◆ [ssh2_publickey_add](#)
- ◆ [ssh2_publickey_init](#)
- ◆ [ssh2_publickey_list](#)
- ◆ [ssh2_publickey_remove](#)
- ◆ [ssh2_scp_recv](#)
- ◆ [ssh2_scp_send](#)
- ◆ [ssh2_sftp](#)
- ◆ [ssh2_sftp_lstat](#)
- ◆ [ssh2_sftp_mkdir](#)
- ◆ [ssh2_sftp_readlink](#)
- ◆ [ssh2_sftp_realpath](#)
- ◆ [ssh2_sftp_rename](#)
- ◆ [ssh2_sftp_rmdir](#)
- ◆ [ssh2_sftp_stat](#)
- ◆ [ssh2_sftp_symlink](#)
- ◆ [ssh2_sftp_unlink](#)
- ◆ [ssh2_shell](#)
- ◆ [ssh2_tunnel](#)
- ◆ [stat](#)
- ◆ [stmt->bind_param](#)
- ◆ [stmt->bind_param](#)
- ◆ [stmt->bind_result](#)
- ◆ [stmt->bind_result](#)
- ◆ [stmt->close_long_data](#)
- ◆ [stmt->data_seek](#)
- ◆ [stmt->data_seek](#)
- ◆ [stmt->execute](#)
- ◆ [stmt->execute](#)
- ◆ [stmt->fetch](#)

- ◆ [stmt->fetch](#)
- ◆ [stmt->free_result](#)
- ◆ [stmt->free_result](#)
- ◆ [stmt->num_rows](#)
- ◆ [stmt->num_rows](#)
- ◆ [stmt->param_count](#)
- ◆ [stmt->param_count](#)
- ◆ [stmt->prepare](#)
- ◆ [stmt->prepare](#)
- ◆ [stmt->reset](#)
- ◆ [stmt->reset](#)
- ◆ [stmt->send_long_data](#)
- ◆ [stmt->send_long_data](#)
- ◆ [str_ireplace](#)
- ◆ [str_pad](#)
- ◆ [str_repeat](#)
- ◆ [str_replace](#)
- ◆ [str_rot13](#)
- ◆ [str_shuffle](#)
- ◆ [str_split](#)
- ◆ [str_word_count](#)
- ◆ [strcasecmp](#)
- ◆ [strchr](#)
- ◆ [strcmp](#)
- ◆ [strcoll](#)
- ◆ [strcspn](#)
- ◆ [stream_bucket_append](#)
- ◆ [stream_bucket_make_writeable](#)
- ◆ [stream_bucket_new](#)
- ◆ [stream_bucket_prepend](#)
- ◆ [stream_context_create](#)
- ◆ [stream_context_get_default](#)
- ◆ [stream_context_get_options](#)
- ◆ [stream_context_set_option](#)
- ◆ [stream_context_set_params](#)
- ◆ [stream_copy_to_stream](#)
- ◆ [stream_filter_append](#)
- ◆ [stream_filter_prepend](#)
- ◆ [stream_filter_register](#)
- ◆ [stream_filter_remove](#)
- ◆ [stream_get_contents](#)
- ◆ [stream_get_filters](#)
- ◆ [stream_get_line](#)
- ◆ [stream_get_meta_data](#)
- ◆ [stream_get_transports](#)
- ◆ [stream_get_wrappers](#)
- ◆ [stream_register_wrapper](#)
- ◆ [stream_select](#)
- ◆ [stream_set_blocking](#)
- ◆ [stream_set_timeout](#)
- ◆ [stream_set_write_buffer](#)
- ◆ [stream_socket_accept](#)
- ◆ [stream_socket_client](#)
- ◆ [stream_socket_enable_crypto](#)

- ◆ [stream_socket_get_name](#)
- ◆ [stream_socket_pair](#)
- ◆ [stream_socket_recvfrom](#)
- ◆ [stream_socket_sendto](#)
- ◆ [stream_socket_server](#)
- ◆ [stream_wrapper_register](#)
- ◆ [stream_wrapper_restore](#)
- ◆ [stream_wrapper_unregister](#)
- ◆ [strftime](#)
- ◆ [strip_tags](#)
- ◆ [stripslashes](#)
- ◆ [stripos](#)
- ◆ [stripslashes](#)
- ◆ [stristr](#)
- ◆ [strlen](#)
- ◆ [strnatcasecmp](#)
- ◆ [strnatcmp](#)
- ◆ [strncasecmp](#)
- ◆ [strncmp](#)
- ◆ [strpbrk](#)
- ◆ [strpos](#)
- ◆ [strptime](#)
- ◆ [strchr](#)
- ◆ [strrev](#)
- ◆ [stripos](#)
- ◆ [strrpos](#)
- ◆ [strspn](#)
- ◆ [strstr](#)
- ◆ [strtok](#)
- ◆ [strtolower](#)
- ◆ [strtotime](#)
- ◆ [strtoupper](#)
- ◆ [strtr](#)
- ◆ [strval](#)
- ◆ [substr](#)
- ◆ [substr_compare](#)
- ◆ [substr_count](#)
- ◆ [substr_replace](#)
- ◆ [swf_actiongeturl](#)
- ◆ [swf_actiongotoframe](#)
- ◆ [swf_actiongotolabel](#)
- ◆ [swf_actionnextframe](#)
- ◆ [swf_actionplay](#)
- ◆ [swf_actionprevframe](#)
- ◆ [swf_actionsettarget](#)
- ◆ [swf_actionstop](#)
- ◆ [swf_actiontogglequality](#)
- ◆ [swf_actionwaitforframe](#)
- ◆ [swf_addbuttonrecord](#)
- ◆ [swf_addcolor](#)
- ◆ [swf_closefile](#)
- ◆ [swf_definebitmap](#)
- ◆ [swf_definefont](#)
- ◆ [swf_defineline](#)

- ◆ [swf_definepoly](#)
- ◆ [swf_definerect](#)
- ◆ [swf_definetext](#)
- ◆ [swf_endbutton](#)
- ◆ [swf_enddoaction](#)
- ◆ [swf_endshape](#)
- ◆ [swf_endsymbol](#)
- ◆ [swf_fontsize](#)
- ◆ [swf_fontslant](#)
- ◆ [swf_fontracking](#)
- ◆ [swf_getbitmapinfo](#)
- ◆ [swf_getfontinfo](#)
- ◆ [swf_getframe](#)
- ◆ [swf_labelframe](#)
- ◆ [swf_lookat](#)
- ◆ [swf_modifyobject](#)
- ◆ [swf_mulcolor](#)
- ◆ [swf_nextid](#)
- ◆ [swf_oncondition](#)
- ◆ [swf_openfile](#)
- ◆ [swf_ortho](#)
- ◆ [swf_ortho2](#)
- ◆ [swf_perspective](#)
- ◆ [swf_placeobject](#)
- ◆ [swf_polarview](#)
- ◆ [swf_popmatrix](#)
- ◆ [swf_posround](#)
- ◆ [swf_pushmatrix](#)
- ◆ [swf_removeobject](#)
- ◆ [swf_rotate](#)
- ◆ [swf_scale](#)
- ◆ [swf_setfont](#)
- ◆ [swf_setframe](#)
- ◆ [swf_shapearc](#)
- ◆ [swf_shapecurveto](#)
- ◆ [swf_shapecurveto3](#)
- ◆ [swf_shapefillbitmapclip](#)
- ◆ [swf_shapefillbitmaptile](#)
- ◆ [swf_shapefilloff](#)
- ◆ [swf_shapefillsolid](#)
- ◆ [swf_shapelinesolid](#)
- ◆ [swf_shapelineto](#)
- ◆ [swf_shapemoveto](#)
- ◆ [swf_showframe](#)
- ◆ [swf_startbutton](#)
- ◆ [swf_startdoaction](#)
- ◆ [swf_startshape](#)
- ◆ [swf_startsymbol](#)
- ◆ [swf_textwidth](#)
- ◆ [swf_translate](#)
- ◆ [swf_viewport](#)
- ◆ [swfaction](#)
- ◆ [swfbitmap](#)
- ◆ [swfbitmap->getheight](#)

- ◆ [swfbitmap->getwidth](#)
- ◆ [swfbutton](#)
- ◆ [swfbutton->addaction](#)
- ◆ [swfbutton->addshape](#)
- ◆ [swfbutton->setaction](#)
- ◆ [swfbutton->setdown](#)
- ◆ [swfbutton->sethit](#)
- ◆ [swfbutton->setover](#)
- ◆ [swfbutton->setup](#)
- ◆ [swfbutton::addasound](#)
- ◆ [swfbutton::setmenu](#)
- ◆ [swfdisplayitem->addcolor](#)
- ◆ [swfdisplayitem->move](#)
- ◆ [swfdisplayitem->moveto](#)
- ◆ [swfdisplayitem->multicolor](#)
- ◆ [swfdisplayitem->remove](#)
- ◆ [swfdisplayitem->rotate](#)
- ◆ [swfdisplayitem->rotateto](#)
- ◆ [swfdisplayitem->scale](#)
- ◆ [swfdisplayitem->scalet](#)
- ◆ [swfdisplayitem->setdepth](#)
- ◆ [swfdisplayitem->setname](#)
- ◆ [swfdisplayitem->setratio](#)
- ◆ [swfdisplayitem->skewx](#)
- ◆ [swfdisplayitem->skewxto](#)
- ◆ [swfdisplayitem->skewy](#)
- ◆ [swfdisplayitem->skewyto](#)
- ◆ [swfdisplayitem::addaction](#)
- ◆ [swfdisplayitem::endmask](#)
- ◆ [swfdisplayitem::getrot](#)
- ◆ [swfdisplayitem::getx](#)
- ◆ [swfdisplayitem::getxscale](#)
- ◆ [swfdisplayitem::getxskew](#)
- ◆ [swfdisplayitem::gety](#)
- ◆ [swfdisplayitem::getyscale](#)
- ◆ [swfdisplayitem::getyskew](#)
- ◆ [swfdisplayitem::setmasklevel](#)
- ◆ [swfdisplayitem::setmatrix](#)
- ◆ [swffill](#)
- ◆ [swffill->moveto](#)
- ◆ [swffill->rotateto](#)
- ◆ [swffill->scalet](#)
- ◆ [swffill->skewxto](#)
- ◆ [swffill->skewyto](#)
- ◆ [swffont](#)
- ◆ [swffont->getwidth](#)
- ◆ [swffont::getascent](#)
- ◆ [swffont::getdescent](#)
- ◆ [swffont::getleading](#)
- ◆ [swffont::getshape](#)
- ◆ [swffont::getutf8width](#)
- ◆ [swffontchar::addchars](#)
- ◆ [swffontchar::addutf8chars](#)
- ◆ [swfgradient](#)

- ◆ [swfgradient->addentry](#)
- ◆ [swfmorph](#)
- ◆ [swfmorph->getshape1](#)
- ◆ [swfmorph->getshape2](#)
- ◆ [swfmovie](#)
- ◆ [swfmovie->add](#)
- ◆ [swfmovie->nextframe](#)
- ◆ [swfmovie->output](#)
- ◆ [swfmovie->remove](#)
- ◆ [swfmovie->save](#)
- ◆ [swfmovie->setbackground](#)
- ◆ [swfmovie->setdimension](#)
- ◆ [swfmovie->setframes](#)
- ◆ [swfmovie->setrate](#)
- ◆ [swfmovie->streammp3](#)
- ◆ [swfmovie::addexport](#)
- ◆ [swfmovie::addfont](#)
- ◆ [swfmovie::importchar](#)
- ◆ [swfmovie::importfont](#)
- ◆ [swfmovie::labelframe](#)
- ◆ [swfmovie::savetofile](#)
- ◆ [swfmovie::startsound](#)
- ◆ [swfmovie::stopsound](#)
- ◆ [swfmovie::writeexports](#)
- ◆ [swfprebuiltclip](#)
- ◆ [swfshape](#)
- ◆ [swfshape->addfill](#)
- ◆ [swfshape->drawcurve](#)
- ◆ [swfshape->drawcurveto](#)
- ◆ [swfshape->drawline](#)
- ◆ [swfshape->drawlineto](#)
- ◆ [swfshape->movepen](#)
- ◆ [swfshape->movepento](#)
- ◆ [swfshape->setleftfill](#)
- ◆ [swfshape->setline](#)
- ◆ [swfshape->setrightfill](#)
- ◆ [swfshape::drawarc](#)
- ◆ [swfshape::drawcircle](#)
- ◆ [swfshape::drawcubic](#)
- ◆ [swfshape::drawcubicto](#)
- ◆ [swfshape::drawglyph](#)
- ◆ [swfsound](#)
- ◆ [swfsoundinstance::loopcount](#)
- ◆ [swfsoundinstance::loopinpoint](#)
- ◆ [swfsoundinstance::loopoutpoint](#)
- ◆ [swfsoundinstance::nomultiple](#)
- ◆ [swfsprite](#)
- ◆ [swfsprite->add](#)
- ◆ [swfsprite->nextframe](#)
- ◆ [swfsprite->remove](#)
- ◆ [swfsprite->setframes](#)
- ◆ [swfsprite::labelframe](#)
- ◆ [swfsprite::startsound](#)
- ◆ [swfsprite::stopsound](#)

- ◆ [swftext](#)
- ◆ [swftext->addstring](#)
- ◆ [swftext->getwidth](#)
- ◆ [swftext->moveto](#)
- ◆ [swftext->setcolor](#)
- ◆ [swftext->setfont](#)
- ◆ [swftext->setheight](#)
- ◆ [swftext->setspacing](#)
- ◆ [swftext::addutf8string](#)
- ◆ [swftext::getascent](#)
- ◆ [swftext::getdescent](#)
- ◆ [swftext::getleading](#)
- ◆ [swftext::getutf8width](#)
- ◆ [swftextfield](#)
- ◆ [swftextfield->addstring](#)
- ◆ [swftextfield->align](#)
- ◆ [swftextfield->setbounds](#)
- ◆ [swftextfield->setcolor](#)
- ◆ [swftextfield->setfont](#)
- ◆ [swftextfield->setheight](#)
- ◆ [swftextfield->setindentation](#)
- ◆ [swftextfield->setleftmargin](#)
- ◆ [swftextfield->setlinespacing](#)
- ◆ [swftextfield->setmargins](#)
- ◆ [swftextfield->setname](#)
- ◆ [swftextfield->setrightmargin](#)
- ◆ [swftextfield::addchars](#)
- ◆ [swftextfield::setpadding](#)
- ◆ [swfvideostream](#)
- ◆ [swfvideostream::getnumframes](#)
- ◆ [swfvideostream::setdimension](#)
- ◆ [sybase_affected_rows](#)
- ◆ [sybase_close](#)
- ◆ [sybase_connect](#)
- ◆ [sybase_data_seek](#)
- ◆ [sybase_deadlock_retry_count](#)
- ◆ [sybase_fetch_array](#)
- ◆ [sybase_fetch_assoc](#)
- ◆ [sybase_fetch_field](#)
- ◆ [sybase_fetch_object](#)
- ◆ [sybase_fetch_row](#)
- ◆ [sybase_field_seek](#)
- ◆ [sybase_free_result](#)
- ◆ [sybase_get_last_message](#)
- ◆ [sybase_min_client_severity](#)
- ◆ [sybase_min_error_severity](#)
- ◆ [sybase_min_message_severity](#)
- ◆ [sybase_min_server_severity](#)
- ◆ [sybase_num_fields](#)
- ◆ [sybase_num_rows](#)
- ◆ [sybase_pconnect](#)
- ◆ [sybase_query](#)
- ◆ [sybase_result](#)
- ◆ [sybase_select_db](#)

- ◆ [sybase_set_message_handler](#)
- ◆ [sybase_unbuffered_query](#)
- ◆ [symlink](#)
- ◆ [syslog](#)
- ◆ [system](#)
- T
- - ◆ [tan](#)
 - ◆ [tanh](#)
 - ◆ [tcpwrap_check](#)
 - ◆ [tempnam](#)
 - ◆ [textdomain](#)
 - ◆ [tidy::construct](#)
 - ◆ [tidy_access_count](#)
 - ◆ [tidy_clean_repair](#)
 - ◆ [tidy_config_count](#)
 - ◆ [tidy_diagnose](#)
 - ◆ [tidy_error_count](#)
 - ◆ [tidy_get_body](#)
 - ◆ [tidy_get_config](#)
 - ◆ [tidy_get_error_buffer](#)
 - ◆ [tidy_get_head](#)
 - ◆ [tidy_get_html](#)
 - ◆ [tidy_get_html_ver](#)
 - ◆ [tidy_get_opt_doc](#)
 - ◆ [tidy_get_output](#)
 - ◆ [tidy_get_release](#)
 - ◆ [tidy_get_root](#)
 - ◆ [tidy_get_status](#)
 - ◆ [tidy_getopt](#)
 - ◆ [tidy_is_xhtml](#)
 - ◆ [tidy_is_xml](#)
 - ◆ [tidy_load_config](#)
 - ◆ [tidy_node->get_attr](#)
 - ◆ [tidy_node->get_nodes](#)
 - ◆ [tidy_node->next](#)
 - ◆ [tidy_node->prev](#)
 - ◆ [tidy_parse_file](#)
 - ◆ [tidy_parse_string](#)
 - ◆ [tidy_repair_file](#)
 - ◆ [tidy_repair_string](#)
 - ◆ [tidy_reset_config](#)
 - ◆ [tidy_save_config](#)
 - ◆ [tidy_set_encoding](#)
 - ◆ [tidy_setopt](#)
 - ◆ [tidy_warning_count](#)
 - ◆ [tidynode->haschildren](#)
 - ◆ [tidynode->hassiblings](#)
 - ◆ [tidynode->isasp](#)
 - ◆ [tidynode->iscomment](#)
 - ◆ [tidynode->ishtml](#)
 - ◆ [tidynode->isjste](#)
 - ◆ [tidynode->isphp](#)
 - ◆ [tidynode->istext](#)
 - ◆ [time](#)

- ◆ [time_nanosleep](#)
- ◆ [time_sleep_until](#)
- ◆ [tmpfile](#)
- ◆ [token_get_all](#)
- ◆ [token_name](#)
- ◆ [touch](#)
- ◆ [trigger_error](#)
- ◆ [trim](#)
- U
- - ◆ [uasort](#)
 - ◆ [ucfirst](#)
 - ◆ [ucwords](#)
 - ◆ [udm_add_search_limit](#)
 - ◆ [udm_alloc_agent](#)
 - ◆ [udm_alloc_agent_array](#)
 - ◆ [udm_api_version](#)
 - ◆ [udm_cat_list](#)
 - ◆ [udm_cat_path](#)
 - ◆ [udm_check_charset](#)
 - ◆ [udm_check_stored](#)
 - ◆ [udm_clear_search_limits](#)
 - ◆ [udm_close_stored](#)
 - ◆ [udm_crc32](#)
 - ◆ [udm_errno](#)
 - ◆ [udm_error](#)
 - ◆ [udm_find](#)
 - ◆ [udm_free_agent](#)
 - ◆ [udm_free_ispell_data](#)
 - ◆ [udm_free_res](#)
 - ◆ [udm_get_doc_count](#)
 - ◆ [udm_get_res_field](#)
 - ◆ [udm_get_res_param](#)
 - ◆ [udm_hash32](#)
 - ◆ [udm_load_ispell_data](#)
 - ◆ [udm_open_stored](#)
 - ◆ [udm_set_agent_param](#)
 - ◆ [uksort](#)
 - ◆ [umask](#)
 - ◆ [uniqid](#)
 - ◆ [unixtojd](#)
 - ◆ [unlink](#)
 - ◆ [unpack](#)
 - ◆ [unregister_tick_function](#)
 - ◆ [unserialize](#)
 - ◆ [unset](#)
 - ◆ [urldecode](#)
 - ◆ [urlencode](#)
 - ◆ [use_soap_error_handler](#)
 - ◆ [user_error](#)
 - ◆ [usleep](#)
 - ◆ [usort](#)
 - ◆ [utf8_decode](#)
 - ◆ [utf8_encode](#)
- V

- - ◆ [var_dump](#)
 - ◆ [var_export](#)
 - ◆ [variant](#)
 - ◆ [variant_abs](#)
 - ◆ [variant_add](#)
 - ◆ [variant_and](#)
 - ◆ [variant_cast](#)
 - ◆ [variant_cat](#)
 - ◆ [variant_cmp](#)
 - ◆ [variant_date from timestamp](#)
 - ◆ [variant_date to timestamp](#)
 - ◆ [variant_div](#)
 - ◆ [variant_eqv](#)
 - ◆ [variant_fix](#)
 - ◆ [variant_get_type](#)
 - ◆ [variant_idiv](#)
 - ◆ [variant_imp](#)
 - ◆ [variant_int](#)
 - ◆ [variant_mod](#)
 - ◆ [variant_mul](#)
 - ◆ [variant_neg](#)
 - ◆ [variant_not](#)
 - ◆ [variant_or](#)
 - ◆ [variant_pow](#)
 - ◆ [variant_round](#)
 - ◆ [variant_set](#)
 - ◆ [variant_set_type](#)
 - ◆ [variant_sub](#)
 - ◆ [variant_xor](#)
 - ◆ [version_compare](#)
 - ◆ [vfprintf](#)
 - ◆ [virtual](#)
 - ◆ [vpopmail_add_alias_domain](#)
 - ◆ [vpopmail_add_alias_domain_ex](#)
 - ◆ [vpopmail_add_domain](#)
 - ◆ [vpopmail_add_domain_ex](#)
 - ◆ [vpopmail_add_user](#)
 - ◆ [vpopmail_alias_add](#)
 - ◆ [vpopmail_alias_del](#)
 - ◆ [vpopmail_alias_del domain](#)
 - ◆ [vpopmail_alias_get](#)
 - ◆ [vpopmail_alias_get all](#)
 - ◆ [vpopmail_auth_user](#)
 - ◆ [vpopmail_del_domain](#)
 - ◆ [vpopmail_del_domain_ex](#)
 - ◆ [vpopmail_del_user](#)
 - ◆ [vpopmail_error](#)
 - ◆ [vpopmail_passwd](#)
 - ◆ [vpopmail_set_user_quota](#)
 - ◆ [vprintf](#)
 - ◆ [vsprintf](#)
- **W**
 - - ◆ [w32api_deftype](#)
 - ◆ [w32api_init_dtype](#)

- ◆ [w32api_invoke_function](#)
- ◆ [w32api_register_function](#)
- ◆ [w32api_set_call_method](#)
- ◆ [wddx_add_vars](#)
- ◆ [wddx_deserialize](#)
- ◆ [wddx_packet_end](#)
- ◆ [wddx_packet_start](#)
- ◆ [wddx_serialize_value](#)
- ◆ [wddx_serialize_vars](#)
- ◆ [wddx_unserialize](#)
- ◆ [wordwrap](#)
- X
- - ◆ [xattr_get](#)
 - ◆ [xattr_list](#)
 - ◆ [xattr_remove](#)
 - ◆ [xattr_set](#)
 - ◆ [xattr_supported](#)
 - ◆ [xdiff_file_diff](#)
 - ◆ [xdiff_file_diff_binary](#)
 - ◆ [xdiff_file_merge3](#)
 - ◆ [xdiff_file_patch](#)
 - ◆ [xdiff_file_patch_binary](#)
 - ◆ [xdiff_string_diff](#)
 - ◆ [xdiff_string_diff_binary](#)
 - ◆ [xdiff_string_merge3](#)
 - ◆ [xdiff_string_patch](#)
 - ◆ [xdiff_string_patch_binary](#)
 - ◆ [xml_error_string](#)
 - ◆ [xml_get_current_byte_index](#)
 - ◆ [xml_get_current_column_number](#)
 - ◆ [xml_get_current_line_number](#)
 - ◆ [xml_get_error_code](#)
 - ◆ [xml_parse](#)
 - ◆ [xml_parse_into_struct](#)
 - ◆ [xml_parser_create](#)
 - ◆ [xml_parser_create_ns](#)
 - ◆ [xml_parser_free](#)
 - ◆ [xml_parser_get_option](#)
 - ◆ [xml_parser_set_option](#)
 - ◆ [xml_set_character_data_handler](#)
 - ◆ [xml_set_default_handler](#)
 - ◆ [xml_set_element_handler](#)
 - ◆ [xml_set_end_namespace_decl_handler](#)
 - ◆ [xml_set_external_entity_ref_handler](#)
 - ◆ [xml_set_notation_decl_handler](#)
 - ◆ [xml_set_object](#)
 - ◆ [xml_set_processing_instruction_handler](#)
 - ◆ [xml_set_start_namespace_decl_handler](#)
 - ◆ [xml_set_unparsed_entity_decl_handler](#)
 - ◆ [xmlrpc_decode](#)
 - ◆ [xmlrpc_decode_request](#)
 - ◆ [xmlrpc_encode](#)
 - ◆ [xmlrpc_encode_request](#)
 - ◆ [xmlrpc_get_type](#)

- ◆ [xmlrpc_is_fault](#)
 - ◆ [xmlrpc_parse_method_descriptions](#)
 - ◆ [xmlrpc_server_add_introspection_data](#)
 - ◆ [xmlrpc_server_call_method](#)
 - ◆ [xmlrpc_server_create](#)
 - ◆ [xmlrpc_server_destroy](#)
 - ◆ [xmlrpc_server_register_introspection_callback](#)
 - ◆ [xmlrpc_server_register_method](#)
 - ◆ [xmlrpc_set_type](#)
 - ◆ [xpath_eval](#)
 - ◆ [xpath_eval_expression](#)
 - ◆ [xpath_new_context](#)
 - ◆ [xpath_register_ns](#)
 - ◆ [xpath_register_ns_auto](#)
 - ◆ [xptr_eval](#)
 - ◆ [xptr_new_context](#)
 - ◆ [xslt_backend_info](#)
 - ◆ [xslt_backend_name](#)
 - ◆ [xslt_backend_version](#)
 - ◆ [xslt_create](#)
 - ◆ [xslt_errno](#)
 - ◆ [xslt_error](#)
 - ◆ [xslt_free](#)
 - ◆ [xslt_getopt](#)
 - ◆ [xslt_process](#)
 - ◆ [xslt_set_base](#)
 - ◆ [xslt_set_encoding](#)
 - ◆ [xslt_set_error_handler](#)
 - ◆ [xslt_set_log](#)
 - ◆ [xslt_set_object](#)
 - ◆ [xslt_set_sax_handler](#)
 - ◆ [xslt_set_sax_handlers](#)
 - ◆ [xslt_set_scheme_handler](#)
 - ◆ [xslt_set_scheme_handlers](#)
 - ◆ [xslt_setopt](#)
 - ◆ [xsltprocessor->_construct\(\)](#)
 - ◆ [xsltprocessor->getparameter\(\)](#)
 - ◆ [xsltprocessor->hasexsltsupport\(\)](#)
 - ◆ [xsltprocessor->importstylesheet\(\)](#)
 - ◆ [xsltprocessor->registerphpfunctions\(\)](#)
 - ◆ [xsltprocessor->removeparameter\(\)](#)
 - ◆ [xsltprocessor->setparameter\(\)](#)
 - ◆ [xsltprocessor->transformtodoc\(\)](#)
 - ◆ [xsltprocessor->transformtouri\(\)](#)
 - ◆ [xsltprocessor->transformtoxml\(\)](#)
- Y
- ◆ [yaz_addinfo](#)
 - ◆ [yaz_ccl_conf](#)
 - ◆ [yaz_ccl_parse](#)
 - ◆ [yaz_close](#)
 - ◆ [yaz_connect](#)
 - ◆ [yaz_database](#)
 - ◆ [yaz_element](#)
 - ◆ [yaz_errno](#)

- ◆ [yaz_error](#)
- ◆ [yaz_es_result](#)
- ◆ [yaz_get_option](#)
- ◆ [yaz_hits](#)
- ◆ [yaz_itemorder](#)
- ◆ [yaz_present](#)
- ◆ [yaz_range](#)
- ◆ [yaz_record](#)
- ◆ [yaz_scan](#)
- ◆ [yaz_scan_result](#)
- ◆ [yaz_schema](#)
- ◆ [yaz_search](#)
- ◆ [yaz_set_option](#)
- ◆ [yaz_sort](#)
- ◆ [yaz_syntax](#)
- ◆ [yaz_wait](#)
- ◆ [yp_all](#)
- ◆ [yp_cat](#)
- ◆ [yp_err_string](#)
- ◆ [yp_errno](#)
- ◆ [yp_first](#)
- ◆ [yp_get_default_domain](#)
- ◆ [yp_master](#)
- ◆ [yp_match](#)
- ◆ [yp_next](#)
- ◆ [yp_order](#)
- Z
- - ◆ [zend_logo_guid](#)
 - ◆ [zend_version](#)
 - ◆ [zip_close](#)
 - ◆ [zip_entry_close](#)
 - ◆ [zip_entry_compressedsize](#)
 - ◆ [zip_entry_compressionmethod](#)
 - ◆ [zip_entry_filesize](#)
 - ◆ [zip_entry_name](#)
 - ◆ [zip_entry_open](#)
 - ◆ [zip_entry_read](#)
 - ◆ [zip_open](#)
 - ◆ [zip_read](#)
 - ◆ [zlib_get_coding_type](#)

13.2 Index des concepts

- A
- "
- ◆ ["ajoute" deux valeurs de variants et retourne le résultat](#)
- A
- - ◆ [Accepte une connexion sur une socket](#)
 - ◆ [Accepte une connexion sur une socket créée par stream_socket_server](#)
 - ◆ [Accomplit une requête aql locale](#)
 - ◆ [Accède à la clé courante](#)
 - ◆ [Accède à la valeur de l'élément courant](#)
 - ◆ [Active l'analyseur rpl](#)

- ◆ Active l'approximation des translations d'objets flash
- ◆ Active l'attribut donné pour les étiquettes des fonction-key (fonctions clés)
- ◆ Active l'interruption de script sur déconnexion du visiteur
- ◆ Active l'option d'alpha blending. pour utiliser les effets de libgd
- ◆ Active l'utilisation de php dans les feuilles de styles xslt
- ◆ Active l'écho d'entrée clavier
- ◆ Active la couche de contrôle des membres et méthodes
- ◆ Active la lecture depuis le maître
- ◆ Active la lecture depuis le maître
- ◆ Active la sauvegarde des paires de remplacement
- ◆ Active la validation automatique oracle
- ◆ Active le déboguer interne de php (php 3)
- ◆ Active le mode auto-validation
- ◆ Active le parseur rpl
- ◆ Active le suivi d'une connexion postgresql
- ◆ Active le traçage comme comportement de menu de bouton
- ◆ Active les attributs suivants
- ◆ Active les couleurs
- ◆ Active ou désactive l'affichage des données de débogage oracle
- ◆ Active ou désactive l'auto-commit des modifications de la base de données
- ◆ Active ou désactive l'entrelacement
- ◆ Active ou désactive la validation automatique
- ◆ Active ou désactive le keypad
- ◆ Active ou désactive le mode auto-commit
- ◆ Active ou désactive le mode autocommit
- ◆ Active ou désactive le mode passif
- ◆ Active ou désactive le rapport interne des fonctions
- ◆ Active ou désactive les alertes frontbase
- ◆ Active ou désactive les fonctions de rapport interne
- ◆ Active ou désactive un code de clé (keycode)
- ◆ Active ou non l'affichage des informations de débogage
- ◆ Active ou non le cryptage sur une socket déjà connectée
- ◆ Active/désactive l'envoi implicite
- ◆ Active/désactive l'option magic_quotes_runtime
- ◆ Active/désactive la bufferisation des lob oracle
- ◆ Active/désactive les informations de méta clé 8-bits
- ◆ Addition de 2 nombres gmp
- ◆ Additionne deux nombres de grande taille
- ◆ Affecte le gestionnaire xml par défaut
- ◆ Affecte les gestionnaires d'entités non déclarées
- ◆ Affecte les gestionnaires d'instructions exécutables
- ◆ Affecte les gestionnaires de début et de fin de balise xml
- ◆ Affecte les gestionnaires de texte littéral
- ◆ Affecte les options d'un analyseur xml
- ◆ Affecte un type à une variable
- ◆ Affiche de nombreuses informations sur php
- ◆ Affiche des informations lisibles pour une variable
- ◆ Affiche des informations sur la connexion au serveur hyperwave
- ◆ Affiche du texte à la position courante
- ◆ Affiche du texte à la prochaine ligne pdf
- ◆ Affiche hw_document
- ◆ Affiche la chaîne sur un terminal dans le mode d'attribut vidéo
- ◆ Affiche la table courant de fonction
- ◆ Affiche le cadre courant

- ◆ Affiche le contenu d'un blob ibase au navigateur
- ◆ Affiche le document pdf qui est en mémoire
- ◆ Affiche le reste du fichier
- ◆ Affiche le résultat sous la forme d'une table html
- ◆ Affiche le texte dans un cadre
- ◆ Affiche le texte à la ligne suivante
- ◆ Affiche le texte à la position courante
- ◆ Affiche le texte à la position courante dans la fenêtre
- ◆ Affiche les crédits de php
- ◆ Affiche les informations d'une variable
- ◆ Affiche ou affecte le paramètre "apache request notes"
- ◆ Affiche tout ce qui est prêt, et rafraîchit l'écran
- ◆ Affiche toutes les données qui restent dans un pointeur gz
- ◆ Affiche un caractère et rafraîchit l'écran
- ◆ Affiche un fichier
- ◆ Affiche un résultat ovrimos sous forme de table html
- ◆ Affiche un texte à une position
- ◆ Affiche un texte à une position donnée
- ◆ Affiche une backtrace
- ◆ Affiche une chaîne de caractères
- ◆ Affiche une chaîne de caractères
- ◆ Affiche une chaîne de caractères formatée
- ◆ Affiche une chaîne formatée
- ◆ Affiche une définition de classe php pour une interface répartissable
- ◆ Affiche une image wbmp
- ◆ Agrège dynamiquement des classes et objets
- ◆ Agrège dynamiquement les méthodes d'une classe à un objet
- ◆ Agrège dynamiquement les propriétés d'une classe à un objet
- ◆ Agrège sélectivement les méthodes d'une classe grâce à une expression rationnelle
- ◆ Agrège sélectivement les méthodes d'une classe grâce à une liste
- ◆ Agrège sélectivement les propriétés d'une classe grâce à une liste
- ◆ Agrège sélectivement les propriétés d'une classe grâce à une expression rationnelle
- ◆ Ajout d'un élément au serveur
- ◆ Ajout initial d'un compartiment au corps
- ◆ Ajout un alias pour un domaine virtuel
- ◆ Ajoute au texte
- ◆ Ajoute cet swfaction à l'instance donnée swfsprite
- ◆ Ajoute des anti-slash dans une chaîne
- ◆ Ajoute des caractères à une police pour exporter une police
- ◆ Ajoute des caractères à une police pour exporter une police
- ◆ Ajoute des caractères à une police qui seront disponibles dans un champ texte
- ◆ Ajoute des données d'introspection
- ◆ Ajoute des données dans un blob ibase fraîchement créé
- ◆ Ajoute des données à la transaction
- ◆ Ajoute des données à un lob oracle
- ◆ Ajoute des slash dans une chaîne, à la mode du langage c
- ◆ Ajoute des variables à un paquet wddx
- ◆ Ajoute différentes limitations de recherche
- ◆ Ajoute du code javascript dans un document fdf
- ◆ Ajoute du code javascript à être exécuté lorsque acrobat ouvre un fdf
- ◆ Ajoute du texte
- ◆ Ajoute dynamiquement une nouvelle méthode à une classe donnée
- ◆ Ajoute dynamiquement une nouvelle méthode à une classe donnée
- ◆ Ajoute la chaîne à la fin des données dans le noeud

- ◆ Ajoute la valeur à la fin
- ◆ Ajoute le mot au dictionnaire personnel
- ◆ Ajoute le mot au dictionnaire personnel de la session courante
- ◆ Ajoute un alias virtuel
- ◆ Ajoute un alias à un domaine virtuel
- ◆ Ajoute un attribut et une valeur à la structure globale
- ◆ Ajoute un attribut à l'entrée courante
- ◆ Ajoute un caractère à la position courante et avance le curseur
- ◆ Ajoute un caractère à la position courante, dans une fenêtre, et avance le curseur
- ◆ Ajoute un compartiment au corps
- ◆ Ajoute un enregistrement dans une base dbase
- ◆ Ajoute un fichier attaché à la page pdf
- ◆ Ajoute un frère à un noeud
- ◆ Ajoute un lien qui démarre un fichier
- ◆ Ajoute un lien web sur la page pdf courante
- ◆ Ajoute un lien à un emplacement web
- ◆ Ajoute un lien à une page dans le même document
- ◆ Ajoute un lien à une page à un second document pdf
- ◆ Ajoute un noeud
- ◆ Ajoute un nouveau domaine virtuel
- ◆ Ajoute un nouveau domaine virtuel
- ◆ Ajoute un nouveau fils à la fin des enfants
- ◆ Ajoute un nouveau fils à la fin des enfants
- ◆ Ajoute un nouveau fils à la fin des fils
- ◆ Ajoute un nouvel attribut
- ◆ Ajoute un nouvel attribut
- ◆ Ajoute un nouvel attribut à l'élément
- ◆ Ajoute un nouvel attribut à l'élément
- ◆ Ajoute un nouvel utilisateur à un domaine virtuel
- ◆ Ajoute un objet dans une animation
- ◆ Ajoute un objet à un sprite
- ◆ Ajoute un paramètre à une procédure stockée mssql (locale ou distante)
- ◆ Ajoute un remplissage plein à la forme flash
- ◆ Ajoute un serveur
- ◆ Ajoute un signet dans la page courante
- ◆ Ajoute un signet à la page courante
- ◆ Ajoute un signet à la page courante
- ◆ Ajoute un template dans le document pdf
- ◆ Ajoute un utilisateur sur la liste des utilisateurs indésirables
- ◆ Ajoute un utilisateur à une base de données de sécurité (uniquement pour ib6 ou plus récent)
- ◆ Ajoute un élément à une collection oracle
- ◆ Ajoute un événement souris dans la queue
- ◆ Ajoute une action au bouton flash
- ◆ Ajoute une annotation
- ◆ Ajoute une annotation dans la page pdf courante
- ◆ Ajoute une annotation de lancement dans la page pdf courante
- ◆ Ajoute une annotation de lien dans la page pdf courante
- ◆ Ajoute une annotation sur un lien vers un fichier dans la page pdf courante
- ◆ Ajoute une chaîne dans une boîte aux lettres
- ◆ Ajoute une chaîne de taille donnée à la position courante
- ◆ Ajoute une chaîne à la position courante
- ◆ Ajoute une chaîne à la position courante
- ◆ Ajoute une chaîne à une nouvelle position dans la fenêtre

- ◆ [Ajoute une clé publique autorisée](#)
- ◆ [Ajoute une contrainte sur une relation](#)
- ◆ [Ajoute une couleur à la liste du gradient](#)
- ◆ [Ajoute une couleur à une transformation](#)
- ◆ [Ajoute une déclaration d'espace de nom à un noeud](#)
- ◆ [Ajoute une entrée dans un dossier ldap](#)
- ◆ [Ajoute une fonction de callback sur une connexion cyrus imap](#)
- ◆ [Ajoute une forme à un bouton](#)
- ◆ [Ajoute une ligne à l'historique](#)
- ◆ [Ajoute une ligne à une relation](#)
- ◆ [Ajoute une miniature sur la page pdf courante](#)
- ◆ [Ajoute une note à la page courante](#)
- ◆ [Ajoute une nouvelle fonction, similaire à `create_function`](#)
- ◆ [Ajoute une ou plusieurs fonctions qui vont gérer les requêtes soap](#)
- ◆ [Ajoute une paire clé/valeur d'une transaction. `replace_transparam\(\)` dépréciée](#)
- ◆ [Ajoute une ressource curl à un curl multiple](#)
- ◆ [Ajoute une règle de réécriture d'url](#)
- ◆ [Ajuste le pointeur de résultat sur une ligne arbitraire dans le résultat](#)
- ◆ [Alias de `aggregate_info`](#)
- ◆ [Alias de `com_get`](#)
- ◆ [Alias de `com_set`](#)
- ◆ [Alias de `com_set`](#)
- ◆ [Alias de `count`](#)
- ◆ [Alias de `current`](#)
- ◆ [Alias de `disk_free_space`](#)
- ◆ [Alias de `exif_read_data`](#)
- ◆ [Alias de `floatval`](#)
- ◆ [Alias de `ftp_close`](#)
- ◆ [Alias de `fwrite`](#)
- ◆ [Alias de `get_included_files`](#)
- ◆ [Alias de `gmp_div_q`](#)
- ◆ [Alias de `gzwrite`](#)
- ◆ [Alias de `highlight_file`](#)
- ◆ [Alias de `imap_headerinfo`](#)
- ◆ [Alias de `imap_list`](#)
- ◆ [Alias de `imap_listscan`](#)
- ◆ [Alias de `imap_lsub`](#)
- ◆ [Alias de `implode`](#)
- ◆ [Alias de `ini_set`](#)
- ◆ [Alias de `is_float`](#)
- ◆ [Alias de `is_float`](#)
- ◆ [Alias de `is_int`](#)
- ◆ [Alias de `is_int`](#)
- ◆ [Alias de `is_writable`](#)
- ◆ [Alias de `ldap_unbind`](#)
- ◆ [Alias de `maxdb_character_set_name`](#)
- ◆ [Alias de `maxdb_options`](#)
- ◆ [Alias de `maxdb_real_escape_string`](#)
- ◆ [Alias de `maxdb_stmt_bind_param`](#)
- ◆ [Alias de `maxdb_stmt_bind_result`](#)
- ◆ [Alias de `maxdb_stmt_close_long_data`](#)
- ◆ [Alias de `maxdb_stmt_execute`](#)
- ◆ [Alias de `maxdb_stmt_fetch`](#)
- ◆ [Alias de `maxdb_stmt_param_count`](#)

- ◆ [Alias de maxdb_stmt_result_metadata](#)
- ◆ [Alias de maxdb_stmt_send_long_data](#)
- ◆ [Alias de mysql_create_db](#)
- ◆ [Alias de mysql_db_query](#)
- ◆ [Alias de mysql_field_flags](#)
- ◆ [Alias de mysql_field_len](#)
- ◆ [Alias de mysql_field_name](#)
- ◆ [Alias de mysql_field_table](#)
- ◆ [Alias de mysql_field_type](#)
- ◆ [Alias de mysql_num_fields](#)
- ◆ [Alias de mysql_num_rows](#)
- ◆ [Alias de mysql_result](#)
- ◆ [Alias de mysql_result](#)
- ◆ [Alias de mysqli_character_set_name](#)
- ◆ [Alias de mysqli_options](#)
- ◆ [Alias de mysqli_real_escape_string](#)
- ◆ [Alias de mysqli_stmt_bind_param](#)
- ◆ [Alias de mysqli_stmt_bind_result](#)
- ◆ [Alias de mysqli_stmt_execute](#)
- ◆ [Alias de mysqli_stmt_fetch](#)
- ◆ [Alias de mysqli_stmt_param_count](#)
- ◆ [Alias de mysqli_stmt_result_metadata](#)
- ◆ [Alias de mysqli_stmt_send_long_data](#)
- ◆ [Alias de oci_bind_by_name](#)
- ◆ [Alias de oci_cancel](#)
- ◆ [Alias de oci_close](#)
- ◆ [Alias de oci_commit](#)
- ◆ [Alias de oci_connect](#)
- ◆ [Alias de oci_define_by_name](#)
- ◆ [Alias de oci_error](#)
- ◆ [Alias de oci_execute](#)
- ◆ [Alias de oci_fetch_all](#)
- ◆ [Alias de oci_fetch](#)
- ◆ [Alias de oci_field_is_null](#)
- ◆ [Alias de oci_field_name](#)
- ◆ [Alias de oci_field_precision](#)
- ◆ [Alias de oci_field_scale](#)
- ◆ [Alias de oci_field_size](#)
- ◆ [Alias de oci_field_type_raw](#)
- ◆ [Alias de oci_field_type](#)
- ◆ [Alias de oci_free_statement](#)
- ◆ [Alias de oci_free_statement](#)
- ◆ [Alias de oci_internal_debug](#)
- ◆ [Alias de oci_lob_export](#)
- ◆ [Alias de oci_lob_import](#)
- ◆ [Alias de oci_new_collection](#)
- ◆ [Alias de oci_new_connect](#)
- ◆ [Alias de oci_new_cursor](#)
- ◆ [Alias de oci_new_descriptor](#)
- ◆ [Alias de oci_num_fields](#)
- ◆ [Alias de oci_num_rows](#)
- ◆ [Alias de oci_parse](#)
- ◆ [Alias de oci_pconnect](#)
- ◆ [Alias de oci_result](#)

- ◆ [Alias de oci_rollback](#)
- ◆ [Alias de oci_server_version](#)
- ◆ [Alias de oci_set_prefetch](#)
- ◆ [Alias de oci_statement_type](#)
- ◆ [Alias de odbc_field_len](#)
- ◆ [Alias de openssl_pkey_get_private](#)
- ◆ [Alias de openssl_pkey_get_public](#)
- ◆ [Alias de recode_string](#)
- ◆ [Alias de rtrim](#)
- ◆ [Alias de session_write_close](#)
- ◆ [Alias de sql_regcase](#)
- ◆ [Alias de sqlite_fetch_single](#)
- ◆ [Alias de stream_get_meta_data](#)
- ◆ [Alias de stream_set_blocking](#)
- ◆ [Alias de stream_set_timeout](#)
- ◆ [Alias de stream_set_write_buffer](#)
- ◆ [Alias de stream_wrapper_register](#)
- ◆ [Alias de strstr](#)
- ◆ [Alias de trigger_error](#)
- ◆ [Alias de wddx_unserialize](#)
- ◆ [Alias de oci-collection->append](#)
- ◆ [Alias de oci-collection->assign](#)
- ◆ [Alias de oci-collection->assignelem](#)
- ◆ [Alias de oci-collection->free](#)
- ◆ [Alias de oci-collection->getelem](#)
- ◆ [Alias de oci-collection->max](#)
- ◆ [Alias de oci-collection->size](#)
- ◆ [Alias de oci-collection->trim](#)
- ◆ [Alias de oci-lob->close](#)
- ◆ [Alias de oci-lob->export](#)
- ◆ [Alias de oci-lob->free](#)
- ◆ [Alias de oci-lob->import](#)
- ◆ [Alias de oci-lob->load](#)
- ◆ [Alias de oci-lob->save](#)
- ◆ [Alias de oci-lob->writetemporary](#)
- ◆ [Alias de addshape\(shape, swfbutton_down\)](#)
- ◆ [Alias de addshape\(shape, swfbutton_hit\)](#)
- ◆ [Alias de addshape\(shape, swfbutton_over\)](#)
- ◆ [Alias de la fonction exit](#)
- ◆ [Alias de swfbutton->addshape\(shape, swfbutton_up\)](#)
- ◆ [Aligne les dessins sur le chemin courant](#)
- ◆ [Alloue de l'espace pour un téléchargement de fichier](#)
- ◆ [Alloue un nouveau curseur oracle](#)
- ◆ [Alloue une couleur pour une image](#)
- ◆ [Alloue une couleur à une image](#)
- ◆ [Alloue une paire de couleur](#)
- ◆ [Alloue une session mnogosearch](#)
- ◆ [Alloue une session mnogosearch](#)
- ◆ [Analyse incrémentalement des données dans un buffer](#)
- ◆ [Analyse la réponse délimitée par virgule alors m_getcell, etc. fonctionnera](#)
- ◆ [Analyse les adresses et retourne un tableau contenant ces données](#)
- ◆ [Analyse les balises d'un fichier ou d'une uri](#)
- ◆ [Analyse les données http get/post/cookie et assigne les variables globales](#)
- ◆ [Analyse un bloc binaire iptc <http://www.iptc.org/> et recherche les balises simples](#)

- ◆ Analyse un certificat x.509 et retourne une ressource
- ◆ Analyse un certificat x509
- ◆ Analyse un document html contenu dans une chaîne
- ◆ Analyse un en-tête mail
- ◆ Analyse un fichier en fonction d'un format
- ◆ Analyse un fichier et retourne une ressource représentant la structure du fichier
- ◆ Analyse une adresse email
- ◆ Analyse une chaîne à l'aide d'un format
- ◆ Analyse une date générée par strftime
- ◆ Analyse une requête http
- ◆ Analyse une requête sql avec oracle
- ◆ Analyse une requête xml est appelle les méthodes associées
- ◆ Analyse une structure xml
- ◆ Analyse une url et retourne ses composants
- ◆ Analyse, exécute et lit le résultat d'une requête oracle
- ◆ Annule
- ◆ Annule complètement une transaction complétée
- ◆ Annule entièrement une autorisation déjà donnée
- ◆ Annule la création d'un blob ibase
- ◆ Annule la fonction exécutée à chaque tick
- ◆ Annule la réécriture d'url
- ◆ Annule la transaction courante
- ◆ Annule la transaction courante
- ◆ Annule les transactions oracle en cours
- ◆ Annule toutes les limitations de recherche mnogosearch
- ◆ Annule un bit
- ◆ Annule une requête asynchrone
- ◆ Annule une requête préparée
- ◆ Annule une transaction
- ◆ Annule une transaction fbsql
- ◆ Annule une transaction ibase
- ◆ Annule une transaction oracle
- ◆ Annule une transaction sans la refermer
- ◆ Annule une transaction sesam
- ◆ Appel une fonction soap
- ◆ Appelle une fonction d'échappement dans les plugins de personnalité msession
- ◆ Appelle l'analyseur ccl
- ◆ Appelle une fonction soap (dépréciée)
- ◆ Appelle une fonction utilisateur
- ◆ Appelle une fonction utilisateur avec les paramètres rassemblés en tableau
- ◆ Appelle une fonction utilisateur à chaque valeur fdf
- ◆ Appelle une fonction windows 32
- ◆ Appelle une méthode d'un composant (déconseillé)
- ◆ Appelle une méthode utilisateur avec un tableau de paramètres
- ◆ Appelle une méthode utilisateur d'un objet
- ◆ Applique l'information d'espacement à la chaîne et l'affiche
- ◆ Applique la syntaxe colorisée à du code php
- ◆ Applique un filtre à une image
- ◆ Applique une correction gamma à l'image
- ◆ Applique une fonction sur les éléments d'un tableau
- ◆ Applique une fonction utilisateur récursivement à chaque membre du tableau

- ◆ Applique une matrice de la convolution 3x3. en utilisant la division de coefficient et l'excentrage
- ◆ Applique une transformation xslt à un objet domdocument
- ◆ Apprête le client pour communication ip/ssl
- ◆ Arc cosinus
- ◆ Arc cosinus hyperbolique
- ◆ Arc sinus
- ◆ Arc sinus hyperbolique
- ◆ Arc tangeant hyperbolique
- ◆ Arc tangent
- ◆ Arc tangent de deux variables
- ◆ Archive un objet
- ◆ Arrondi
- ◆ Arrondit au nombre supérieur
- ◆ Arrondit le variant au nombre spécifié de décimales
- ◆ Arrondit à l'entier inférieur
- ◆ Arrête l'animation flash
- ◆ Arrête l'exécution durant quelques secondes
- ◆ Arrête la lecture de la source
- ◆ Assigne l'action du bouton
- ◆ Assigne la couleur de terminal pour l'index -1
- ◆ Assigne un nom à un jeu de résultats ibase
- ◆ Assigne une nouvelle valeur pour un objet variant
- ◆ Assigne une valeur à un élément d'une collection oracle
- ◆ Assigne une valeur à une collection depuis une autre collection oracle
- ◆ Associe un son avec une transition de bouton
- ◆ Associe une valeur à un paramètre
- ◆ Associe une variable php à une variable sql oracle
- ◆ Assurez-vous qu'il n'y avait aucune falsification de données dans le message de confirmation reçu
- ◆ Assèche des données
- ◆ Attache le dessin au chemin courant
- ◆ Attache un attribut binaire
- ◆ Attache un attribut binaire à un vendeur spécifique
- ◆ Attache un attribut chaîne de caractères
- ◆ Attache un attribut entier
- ◆ Attache un attribut entier à un vendeur spécifique
- ◆ Attache un attribut ip-address spécifique à un vendeur
- ◆ Attache un attribut sous la forme d'une chaîne à un vendeur spécifique
- ◆ Attache un filtre à un flux en début de liste
- ◆ Attache un filtre à un flux en fin de liste
- ◆ Attache une adresse ip en tant qu'attribut
- ◆ Attend l'exécution d'une requête
- ◆ Attend la fin de l'exécution d'un processus fils
- ◆ Attend ou retourne le statut d'un processus fils
- ◆ Attend un événement ibase
- ◆ Attend une connexion sur une socket
- ◆ Attend x micro secondes
- ◆ Attendre pendant un nombre de secondes et de nanosecondes
- ◆ Attributs de l'objet dans l'ancrage
- ◆ Attributs des ancrages d'un document
- ◆ Attributs des documents fils d'un groupe
- ◆ Attributs des parents
- ◆ Authentification au serveur ldap

- ◆ Authentification au serveur ldap en utilisant sasl
- ◆ Avance d'un cadre
- ◆ Avance le pointeur interne d'un tableau
- ◆ Avance à la prochaine ligne de résultats d'un gestionnaire de lignes de résultats multiples
- ◆ Avance à une position
- **B**
-
- **C**
-
- ◆ Balise de corps d'un document
- ◆ Calcul des statistiques sur une table
- ◆ Calcul l'intersection de deux tableaux avec des tests sur l'index, compare les données et les indexes des deux tableaux en utilisant une fonction de callback
- ◆ Calcul l'intersection de deux tableaux avec des tests sur l'index, compare les donnée en utilisant une fonction de callback
- ◆ Calcul l'intersection de deux tableaux avec des tests sur les index, compare les index en utilisant une fonction de callback
- ◆ Calcul l'intersection de deux tableaux en utilisant les clés pour comparaison
- ◆ Calcul l'intersection de deux tableaux en utilisant une fonction de callback sur les clés pour comparaison
- ◆ Calcul l'intersection de deux tableaux, compare les données en utilisant une fonction de callback
- ◆ Calcul la différence de deux tableaux en utilisant les clés pour comparaison
- ◆ Calcul la longueur de l'hypoténuse d'un triangle à angle droit
- ◆ Calcul un chemin xptr à partir d'une chaîne
- ◆ Calcule l'intersection de deux tableaux avec des tests sur les index
- ◆ Calcule l'intersection de tableaux
- ◆ Calcule la clé metaphone
- ◆ Calcule la clé soundex
- ◆ Calcule la différence de deux tableaux associatifs, compare les données et les index avec une fonction de callback
- ◆ Calcule la différence de deux tableaux, en prenant en compte les clés
- ◆ Calcule la différence entre des tableaux avec vérification des index, compare les données avec une fonction de callback
- ◆ Calcule la différence entre deux tableaux
- ◆ Calcule la différence entre deux tableaux associatifs, à l'aide d'une fonction utilisateur
- ◆ Calcule la différence entre deux tableaux en utilisant une fonction callback
- ◆ Calcule la différence entre deux tableaux en utilisant une fonction de callback sur les clés pour comparaison
- ◆ Calcule la distance levenshtein entre deux chaînes
- ◆ Calcule la largeur de la chaîne de caractères donnée dans cette police à ca pleine taille
- ◆ Calcule la largeur de la chaîne de caractères donnée dans les objets de texte de police et taille courante
- ◆ Calcule la longueur d'une chaîne
- ◆ Calcule la similarité de deux chaînes
- ◆ Calcule la somme de contrôle crc32
- ◆ Calcule la somme des valeurs du tableau
- ◆ Calcule la taille d'une chaîne
- ◆ Calcule le complémentaire d'un nombre
- ◆ Calcule le hachage demandé par ezmlm
- ◆ Calcule le md5 d'un fichier
- ◆ Calcule le md5 d'une chaîne
- ◆ Calcule le nombre de couleurs d'une palette

- ◆ Calcule le produit des valeurs du tableau
- ◆ Calcule le rectangle d'encadrement pour un texte, en utilisant la police courante et freetype2
- ◆ Calcule le reste modulo d'un nombre élevé à une puissance
- ◆ Calcule le sha1 d'un fichier
- ◆ Calcule le sha1 d'une chaîne de caractères
- ◆ Calcule un chemin xpath à partir d'une chaîne
- ◆ Calcule un chemin xpath à partir d'une chaîne
- ◆ Calcule un hash
- ◆ Carré parfait gmp
- ◆ Cesse l'utilisation de l'attribut 'standout'
- ◆ Cesse l'utilisation de ncurses et efface l'écran
- ◆ Change de dossier
- ◆ Change de dossier et passe au dossier parent
- ◆ Change de nom sur le serveur
- ◆ Change dynamiquement le code de la méthode donnée
- ◆ Change dynamiquement le code de la méthode donnée
- ◆ Change dynamiquement le nom d'une méthode donnée
- ◆ Change dynamiquement le nom de la méthode donnée
- ◆ Change l'espacement des caractères
- ◆ Change l'inclinaison de la police courante
- ◆ Change l'utilisateur de la connexion spécifiée
- ◆ Change la bufferisation d'entrée
- ◆ Change la casse des clés du tableau
- ◆ Change la couleur dans une palette à l'index donné
- ◆ Change la police courante
- ◆ Change la position courante
- ◆ Change la priorité d'exécution du processus courant
- ◆ Change la priorité d'un processus
- ◆ Change la taille de la police
- ◆ Change le "umask" courant
- ◆ Change le codage vectoriel d'un caractère dans une police
- ◆ Change le dossier ftp courant
- ◆ Change le dossier racine
- ◆ Change le groupe propriétaire d'un fichier
- ◆ Change le groupe propriétaire du fichier
- ◆ Change le mode de suggestion
- ◆ Change le mode du fichier
- ◆ Change le mot de passe de l'utilisateur spécifié
- ◆ Change le nom d'une fonction
- ◆ Change le nom d'utilisateur de la session active
- ◆ Change le nom de l'utilisateur de la connexion active
- ◆ Change les attributs d'un objet (obsolète)
- ◆ Charge du html à partir d'un fichier
- ◆ Charge du html à partir d'une chaîne de caractères
- ◆ Charge du xml depuis un fichier
- ◆ Charge du xml depuis une chaîne de caractères
- ◆ Charge les données ispell
- ◆ Charge un buffer avec des données
- ◆ Charge un fichier .wav dans le buffer
- ◆ Charge un fichier de configuration ascii tidy avec l'encodage spécifié
- ◆ Charge un fichier sur un serveur ftp
- ◆ Charge un fichier sur un serveur ftp
- ◆ Charge un module dotnet

- ◆ [Charge un nouveau dictionnaire](#)
- ◆ [Charge un nouveau dictionnaire](#)
- ◆ [Charge un nouveau dictionnaire \[obsolète\]](#)
- ◆ [Charge un nouveau dictionnaire avec un dictionnaire personnel](#)
- ◆ [Charge un typelib](#)
- ◆ [Charge une extension php à la volée](#)
- ◆ [Charge une nouvelle police](#)
- ◆ [Charge une police](#)
- ◆ [Charge une police](#)
- ◆ [Charge une police postscript type 1 depuis un fichier](#)
- ◆ [Chemin du dossier de travail courant](#)
- ◆ [Cherche les notes par mots-clés dans database_name](#)
- ◆ [Cherche tous les éléments qui ont le nom de balise donné](#)
- ◆ [Cherche un élément avec un certain identifiant](#)
- ◆ [Chiffre indéchiffrable \(hashing\)](#)
- ◆ [Chiffre des données avec une clé privée](#)
- ◆ [Chiffre des données avec une clé publique](#)
- ◆ [Chiffre les données](#)
- ◆ [Chiffre un message s/mime](#)
- ◆ [Chiffre un texte](#)
- ◆ [Chiffre/déchiffre des données en mode cbc](#)
- ◆ [Chiffre/déchiffre des données en mode cfb](#)
- ◆ [Chiffre/déchiffre des données en mode ofb](#)
- ◆ [Choisi une nouvelle position et dessine une ligne horizontale avec le caractère donné, et de taille maximale donnée](#)
- ◆ [Choisit l'encodage du client postgresql](#)
- ◆ [Choisit la couleur rgb de remplissage et de dessin](#)
- ◆ [Choisit la couleur utilisée pour le dessin](#)
- ◆ [Choisit la couleur utilisée pour le remplissage](#)
- ◆ [Choisit le fichier qui contient le dictionnaire personnel](#)
- ◆ [Choisit le fichier qui contient les paires de remplacement](#)
- ◆ [Choisit le mode de remplissage flash par texture](#)
- ◆ [Choisit le mode de remplissage par texture flash répétée](#)
- ◆ [Choisit le mode par défaut des objets blob pour toutes les requêtes select](#)
- ◆ [Choisit le mode par défaut des objets byte](#)
- ◆ [Choisit le mode par défaut des objets text](#)
- ◆ [Choisit le niveau de qualité de l'animation flash](#)
- ◆ [Choisit le point de départ de la recherche par expression rationnelle](#)
- ◆ [Choisit le style de bord autour des liens et annotations](#)
- ◆ [Choisit un niveau de gris comme couleur de dessin](#)
- ◆ [Choisit une couleur rgb comme couleur de dessin](#)
- ◆ [Choisit une couleur rgb comme couleur de dessin et de remplissage](#)
- ◆ [Choisit une couleur rgb comme couleur de remplissage](#)
- ◆ [Classe anti-sandbox runkit](#)
- ◆ [Classe com](#)
- ◆ [Classe de dossier dir](#)
- ◆ [Classe runkit sandbox -- machine virtuelle php](#)
- ◆ [Classe variant](#)
- ◆ [Clone un noeud](#)
- ◆ [Clone un noeud](#)
- ◆ [Clone un objet](#)
- ◆ [Clôt un paquet wddx](#)
- ◆ [Colorisation syntaxique d'un fichier](#)
- ◆ [Combine plusieurs tableaux ensemble, récursivement](#)

- ◆ Commence l'analyse d'un document xml
- ◆ Commence l'utilisation de l'attribut 'standout'
- ◆ Commence la description d'une liste d'actions pour le cadre courant
- ◆ Commence la définition d'un bouton
- ◆ Commence un nouveau chemin
- ◆ Commence un nouveau document
- ◆ Commence un nouveau paquet wddx avec une structure
- ◆ Commence une forme complexe
- ◆ Commence une nouvelle page
- ◆ Commence une nouvelle page
- ◆ Compacte des données dans une chaîne binaire
- ◆ Compacte une base dbase
- ◆ Comparaison binaire de chaînes
- ◆ Comparaison binaire des n premiers caractères
- ◆ Comparaison de chaînes avec l'algorithme d'"ordre naturel"
- ◆ Comparaison de chaînes avec l'algorithme d'"ordre naturel" (insensible à la casse)
- ◆ Comparaison de chaînes localisées
- ◆ Comparaison insensible à la casse de chaînes binaires
- ◆ Compare des nombres gmp
- ◆ Compare deux chaînes de versions au format des versions php
- ◆ Compare deux chaînes depuis un offset jusqu'à une longueur en caractères
- ◆ Compare deux dates
- ◆ Compare deux lignes dbx afin de les trier
- ◆ Compare deux lob/file oracle
- ◆ Compare deux nombres de grande taille
- ◆ Compare deux variants
- ◆ Compare en binaire des chaînes de caractères
- ◆ Compare une entrée avec des valeurs d'attributs
- ◆ Compile un fichier de code php et retourne le tableau d'opcode résultant
- ◆ Compile une chaîne de code php et retourne le tableau d'opcode résultant
- ◆ Complète la définition de la forme flash courante
- ◆ Complète un tableau avec une valeur jusqu'à la longueur spécifiée
- ◆ Complète une chaîne jusqu'à une taille donnée
- ◆ Compresse une chaîne
- ◆ Compresse une chaîne
- ◆ Compresse une chaîne avec bzip2
- ◆ Compression lzf
- ◆ Compte de nombre d'éléments dans un itérateur
- ◆ Compte de population
- ◆ Compte le nombre d'enregistrements dans une base dbase
- ◆ Compte le nombre d'entrées après une recherche
- ◆ Compte le nombre d'occurrences d'une sous-chaîne
- ◆ Compte le nombre d'occurrences de segments dans une chaîne
- ◆ Compte le nombre d'éléments d'un tableau ou le nombre de propriétés d'un objet
- ◆ Compte le nombre de champs d'une base dbase
- ◆ Compte le nombre de lignes déjà lues dans un résultat
- ◆ Compte le nombre de mots utilisés dans une chaîne
- ◆ Compte le nombre de sessions
- ◆ Compte le nombre de valeurs dans un tableau
- ◆ Concatène deux valeurs variantes ensemble et retourne le résultat
- ◆ Concatène une matrice avec le ctm
- ◆ Configure l'analyseur ccl
- ◆ Configure l'enregistrement des informations complètes du canal alpha lors de sauvegardes d'images png

- ◆ [Configure l'identifiant vers lequel les liens sont calculés](#)
- ◆ [Configure l'inclinaison d'un texte](#)
- ◆ [Configure l'uri de base de toutes les transformations xslt](#)
- ◆ [Configure l'échelle \(??\)](#)
- ◆ [Configure l'échelle du document](#)
- ◆ [Configure la "miter limit"](#)
- ◆ [Configure la bufferisation de fichier pour un flux](#)
- ◆ [Configure la connexion à l'imprimante](#)
- ◆ [Configure la couleur de dessin et de remplissage](#)
- ◆ [Configure la couleur de dessin et de remplissage à un niveau de gris](#)
- ◆ [Configure la couleur de dessin à un niveau de gris](#)
- ◆ [Configure la couleur de remplissage à un niveau de gris](#)
- ◆ [Configure la couleur des bords autour des liens et annotations](#)
- ◆ [Configure la couleur des clés soft label](#)
- ◆ [Configure la durée d'expiration d'un flux](#)
- ◆ [Configure la largeur de ligne](#)
- ◆ [Configure la matrice de transformation](#)
- ◆ [Configure la police courante](#)
- ◆ [Configure la position à plat \(flatness\)](#)
- ◆ [Configure la rotation](#)
- ◆ [Configure le cadre de destination pour l'affichage du formulaire](#)
- ◆ [Configure le cadre fdf de destination pour le formulaire](#)
- ◆ [Configure le contexte de flux pour la prochaine opération libxml](#)
- ◆ [Configure le délai d'attente d'une base sqlite occupée](#)
- ◆ [Configure le fichier d'historique pour les messages xslt](#)
- ◆ [Configure le fichier de remplacement de polices externes](#)
- ◆ [Configure le gestionnaire d'erreurs du processeur xslt](#)
- ◆ [Configure le gestionnaire de caractères](#)
- ◆ [Configure le gestionnaire de messages sybase](#)
- ◆ [Configure le gestionnaire xml de données](#)
- ◆ [Configure le gestionnaire xml de notations](#)
- ◆ [Configure le gestionnaire xml de références externes](#)
- ◆ [Configure le jeu de caractères pour l'analyse des documents xml](#)
- ◆ [Configure le journal de connexion](#)
- ◆ [Configure le mode bloquant d'un flux](#)
- ◆ [Configure le mode de pointillé](#)
- ◆ [Configure le nombre de tentatives lors de blocages \(deadlock\)](#)
- ◆ [Configure le paramètre de linecap](#)
- ◆ [Configure le paramètre de linejoin](#)
- ◆ [Configure le style des lignes autour des liens et annotations](#)
- ◆ [Configure le style pour le dessin des lignes](#)
- ◆ [Configure le temps d'expiration des séquences clavier spéciales](#)
- ◆ [Configure le verrouillage et l'isolation de transactions](#)
- ◆ [Configure les chaînes et les expressions rationnelles pour le support des caractères multi-octets](#)
- ◆ [Configure les critères de tri](#)
- ◆ [Configure les délais entre les clics de souris](#)
- ◆ [Configure les fonctions de stockage de sessions](#)
- ◆ [Configure les gestionnaires de schéma du processeur xslt](#)
- ◆ [Configure les gestionnaires sax qui seront appelés pour gérer les documents xml](#)
- ◆ [Configure les paramètres pour un flux/gestionnaire/contexte](#)
- ◆ [Configure les événements de souris à surveiller](#)
- ◆ [Configure ou retourne le timeout](#)
- ◆ [Configure un gestionnaire de schéma pour un processeur xslt](#)

- ◆ [Configure un objet snmp](#)
- ◆ [Configure un écran depuis un fichier de sauvegarde](#)
- ◆ [Configure une action à exécuter lorsque la connexion irc se termine](#)
- ◆ [Configure une fonction de callback pour refaire des liaisons lors de recherche de référents](#)
- ◆ [Configure une nouvelle valeur rgb pour une couleur](#)
- ◆ [Configure une option pour un flux/gestionnaire/contexte](#)
- ◆ [Connecte des événements d'un objet com sur un objet php](#)
- ◆ [Connecte à un serveur irc](#)
- ◆ [Connexion au service de gestion interbase](#)
- ◆ [Connexion à un serveur](#)
- ◆ [Connexion à un serveur ssh](#)
- ◆ [Connexion à une source](#)
- ◆ [Considère deux mots accolés comme un composé](#)
- ◆ [Constructeur soapclient](#)
- ◆ [Constructeur soapfault](#)
- ◆ [Constructeur soapheader](#)
- ◆ [Constructeur soapparam](#)
- ◆ [Constructeur soapserver](#)
- ◆ [Constructeur soapvar](#)
- ◆ [Construit un en-tête mime avec les champs field_name et field_value](#)
- ◆ [Construit un nouvel itérateur de dossier à partir d'un chemin](#)
- ◆ [Construit un nouvel objet tableau](#)
- ◆ [Construit un nouvel objet tidy](#)
- ◆ [Construit un objet simplexmlelement à partir d'une node dom](#)
- ◆ [Consulte un paramètre de configuration courant du serveur](#)
- ◆ [Contenu d'un document](#)
- ◆ [Continue un texte à la ligne suivante](#)
- ◆ [Contrôle l'utilisation des informations d'environnement concernant la taille du terminal](#)
- ◆ [Contrôle l'utilisation des noms étendus dans les descriptions terminfo](#)
- ◆ [Contrôle la situation, l'apparence et la zone active du bouton flash courant](#)
- ◆ [Conversion d'encodage](#)
- ◆ [Conversion d'octal en décimal](#)
- ◆ [Conversion de radians en degrés](#)
- ◆ [Convertit de binaire en décimal](#)
- ◆ [Convertit de décimal en binaire](#)
- ◆ [Convertit de décimal en hexadécimal](#)
- ◆ [Convertit de décimal en octal](#)
- ◆ [Convertit de hexadécimal en décimal](#)
- ◆ [Convertit des données binaires en représentation hexadécimale](#)
- ◆ [Convertit des données brutes en adresse ip](#)
- ◆ [Convertit des données brutes en chaîne de caractères](#)
- ◆ [Convertit des données brutes en entier](#)
- ◆ [Convertit des tableaux associatifs en une commande postgresql](#)
- ◆ [Convertit des variants en valeurs entières, et performe alors une division](#)
- ◆ [Convertit du texte au format mime en utf8](#)
- ◆ [Convertit entre les différents "kana"](#)
- ◆ [Convertit l'arbre xml interne en une chaîne de caractères](#)
- ◆ [Convertit l'arbre xml interne en une chaîne de caractères au format html](#)
- ◆ [Convertit l'encodage de variables](#)
- ◆ [Convertit le guid binaire de 16 octets en une chaîne de caractères](#)
- ◆ [Convertit le nombre de jours du calendrier julien en date grégorienne](#)
- ◆ [Convertit le nombre de jours du calendrier julien en date du calendrier français](#)

républicain

- ◆ Convertit le nombre de jours du calendrier julien en date du calendrier juif
- ◆ Convertit le nombre de jours du calendrier julien en date du calendrier julien
- ◆ Convertit le nombre de jours julien en un calendrier et retourne des détails
- ◆ Convertit les attributs d'un objet en tableau
- ◆ Convertit les caractères 8859 en caractères t61
- ◆ Convertit les caractères spéciaux en entités html
- ◆ Convertit les caractères t6 en caractères 8859
- ◆ Convertit les entités html spéciales en caractères
- ◆ Convertit les nouvelles lignes et retours chariot en sauts de ligne
- ◆ Convertit tous les caractères éligibles en entités html
- ◆ Convertit toutes les entités html en caractères normaux
- ◆ Convertit un calendrier en nombre de jours julien
- ◆ Convertit un chemin et un identifiant de projet en une clé system v ipc
- ◆ Convertit un dn en format ufn (user friendly naming)
- ◆ Convertit un document dom xml en fichier
- ◆ Convertit un fichier xml en objet
- ◆ Convertit un jours du calendrier julien en un nombre de jours du calendrier julien
- ◆ Convertit un jours julien en timestamp unix
- ◆ Convertit un noeud en chaîne
- ◆ Convertit un nombre de degrés en radians
- ◆ Convertit un nombre entre des bases arbitraires
- ◆ Convertit un nombre gmp en chaîne
- ◆ Convertit un nombre gmp en entier
- ◆ Convertit un numéro d'erreur ldap en message d'erreur
- ◆ Convertit un paquet d'adresses internet en une représentation humainement lisible
- ◆ Convertit un tableau en un objet
- ◆ Convertit un texte logique hébreux en texte visuel
- ◆ Convertit un texte logique hébreux en texte visuel, avec retours à la ligne
- ◆ Convertit un timestamp en une chaîne de caractères
- ◆ Convertit un timestamp unix en un jours julien
- ◆ Convertit un variant en un autre type "sur-place"
- ◆ Convertit un variant en un nouvel objet variant de type différent
- ◆ Convertit une adresse ip (ipv4) en adresse ip numérique
- ◆ Convertit une adresse ip lisible en sa représentation in_addr
- ◆ Convertit une chaîne contenant une adresse (ipv4) ip numérique en adresse littérale
- ◆ Convertit une chaîne d'un jeu de caractères cyrillique à l'autre
- ◆ Convertit une chaîne dans un jeu de caractères
- ◆ Convertit une chaîne de caractères en tableau
- ◆ Convertit une chaîne en nombre à virgule flottante
- ◆ Convertit une chaîne iso-8859-1 en texte utf-7 modifié
- ◆ Convertit une chaîne iso-8859-1 en utf-8
- ◆ Convertit une chaîne logique en chaîne visuelle
- ◆ Convertit une chaîne quoted-printable en chaîne 8 bits
- ◆ Convertit une chaîne utf-8 en iso-8859-1
- ◆ Convertit une chaîne xml en objet
- ◆ Convertit une chaîne à 8 bits en une chaîne à guillemets
- ◆ Convertit une chaîne à 8 bits en une chaîne à base64
- ◆ Convertit une chaîne à guillemets en une chaîne à 8 bits
- ◆ Convertit une classe de base à une classe héritée, ajoute une méthode ancestrale lorsque approprié
- ◆ Convertit une classe héritée à une classe de base, supprime toute méthode pour qui la portée est ancestrale
- ◆ Convertit une date du calendrier français républicain en nombre de jours du

calendrier julien

- ◆ Convertit une date du calendrier juif en nombre de jours du calendrier julien
- ◆ Convertit une date grégorienne en nombre de jours du calendrier julien
- ◆ Convertit une image en vraies couleurs en image à palette
- ◆ Convertit une image jpeg en image wbmp
- ◆ Convertit une image png en image wbmp
- ◆ Convertit une valeur date/temps variante en un timestamp unix
- ◆ Copie des données depuis un flux vers un autre
- ◆ Copie des objets hyperwave
- ◆ Copie et fusionne une partie d'une image
- ◆ Copie et fusionne une partie d'une image en niveaux de gris
- ◆ Copie et redimensionne une partie d'une image
- ◆ Copie la fenêtre dans l'écran virtuel
- ◆ Copie la palette d'une image à l'autre
- ◆ Copie les clés soft label à l'écran
- ◆ Copie les messages spécifiés dans une boîte aux lettres
- ◆ Copie physique
- ◆ Copie toutes les clés soft label à l'écran virtuel
- ◆ Copie un fichier
- ◆ Copie un itérateur dans un tableau
- ◆ Copie un lob oracle
- ◆ Copie une base de données lotus notes
- ◆ Copie une fonction vers un nom de fonction nouveau
- ◆ Copie une méthode d'une classe vers une autre classe
- ◆ Copie une méthode d'une classe à une autre
- ◆ Copie une partie d'une image
- ◆ Copie une ressource curl avec toutes ses préférences
- ◆ Copie une région depuis un pad dans l'écran virtuel
- ◆ Copie une région depuis un pad dans un écran virtuel
- ◆ Copie une table dans un tableau
- ◆ Copie, redimensionne, rééchantillonne une image
- ◆ Cosinus
- ◆ Cosinus hyperbolic
- ◆ Coupe le noeud en deux noeuds à l'endroit spécifié
- ◆ Coupe une chaîne en segments
- ◆ Coupe une chaîne en segments
- ◆ Coupe une partie de chaîne
- ◆ Création d'un analyseur xml
- ◆ Crée et initialise une structure mcve_conn
- ◆ Crée le lien depuis la source vers le destinataire, relativement à rootid
- ◆ Crée ou ouvre un bloc de mémoire partagée
- ◆ Crée ou ouvre un segment de mémoire partagée
- ◆ Crée ou s'attache à une file de messages
- ◆ Crée un analyseur xml
- ◆ Crée un arbre d'objets php à partir d'un document xml
- ◆ Crée un blob
- ◆ Crée un blob ibase pour ajouter des données
- ◆ Crée un blob ibase, y copie un fichier et le referme
- ◆ Crée un clob
- ◆ Crée un contexte d'imprimante
- ◆ Crée un contexte de flux
- ◆ Crée un contexte de traitement audio
- ◆ Crée un document pdf pour y afficher des données fdf
- ◆ Crée un document xml vide

- ◆ Crée un dossier
- ◆ Crée un dossier
- ◆ Crée un dossier sur un serveur ftp
- ◆ Crée un espace d'exécution pour une application dans un dossier virtuel
- ◆ Crée un fichier avec un nom unique
- ◆ Crée un fichier fifo (first in, first out) (un pipe nommé)
- ◆ Crée un fichier spécial ou ordinaire (posix.1)
- ◆ Crée un fichier temporaire
- ◆ Crée un lien
- ◆ Crée un lien avec un objet
- ◆ Crée un lien symbolique
- ◆ Crée un lien symbolique
- ◆ Crée un message mime
- ◆ Crée un morphing
- ◆ Crée un motif basé sur le ton
- ◆ Crée un nom de navigateur dans database_name
- ◆ Crée un nombre gmp
- ◆ Crée un nouveau bouton
- ◆ Crée un nouveau calendrier
- ◆ Crée un nouveau calendrier
- ◆ Crée un nouveau champ texte
- ◆ Crée un nouveau compartiment pour l'utiliser sur le flux courant
- ◆ Crée un nouveau conteneur et l'associe à une fenêtre
- ◆ Crée un nouveau contexte xpath
- ◆ Crée un nouveau contexte xpath
- ◆ Crée un nouveau document
- ◆ Crée un nouveau document fdf
- ◆ Crée un nouveau fragment de document
- ◆ Crée un nouveau noeud
- ◆ Crée un nouveau noeud avec un espace de noms associé
- ◆ Crée un nouveau noeud cdata
- ◆ Crée un nouveau noeud cdata
- ◆ Crée un nouveau noeud de commentaire
- ◆ Crée un nouveau noeud de commentaire
- ◆ Crée un nouveau noeud de référence d'entité
- ◆ Crée un nouveau noeud de texte
- ◆ Crée un nouveau noeud de texte
- ◆ Crée un nouveau noeud de traitement d'instructions (pi)
- ◆ Crée un nouveau noeud pi
- ◆ Crée un nouveau noeud élément
- ◆ Crée un nouveau pad (fenêtre)
- ◆ Crée un nouveau serveur web virtuel
- ◆ Crée un nouveau stylo
- ◆ Crée un nouvel analyseur xslt
- ◆ Crée un nouvel attribut
- ◆ Crée un nouvel attribut
- ◆ Crée un nouvel attribut avec un espace de noms associé
- ◆ Crée un nouvel index secondaire pour une relation
- ◆ Crée un nouvel objet document postscript
- ◆ Crée un nouvel objet domattr
- ◆ Crée un nouvel objet domcomment
- ◆ Crée un nouvel objet domdocument
- ◆ Crée un nouvel objet domelement
- ◆ Crée un nouvel objet dumentityreference

- ◆ [Crée un nouvel objet domimplementation](#)
- ◆ [Crée un nouvel objet domprocessinginstruction](#)
- ◆ [Crée un nouvel objet domtext](#)
- ◆ [Crée un nouvel objet domxpath](#)
- ◆ [Crée un nouvel objet paradox](#)
- ◆ [Crée un nouvel objet pdf](#)
- ◆ [Crée un nouvel objet texte](#)
- ◆ [Crée un nouvel objet xsltprocessor](#)
- ◆ [Crée un nouvel élément avec un espace de noms associé](#)
- ◆ [Crée un objet 'animation'](#)
- ◆ [Crée un objet bitmap](#)
- ◆ [Crée un objet blob](#)
- ◆ [Crée un objet char](#)
- ◆ [Crée un objet de grande taille postgresql](#)
- ◆ [Crée un objet de remplissage](#)
- ◆ [Crée un objet dom document du type spécifié avec ses éléments](#)
- ◆ [Crée un objet dom pour un document xml](#)
- ◆ [Crée un objet dom à partir d'un fichier xml](#)
- ◆ [Crée un objet domdocumenttype vide](#)
- ◆ [Crée un objet domxsltstylesheet à partir d'un document xsl dans un fichier](#)
- ◆ [Crée un objet domxsltstylesheet à partir d'un document xsl dans une chaîne](#)
- ◆ [Crée un objet domxsltstylesheet à partir d'un objet domdocument](#)
- ◆ [Crée un objet gradient](#)
- ◆ [Crée un objet slob et l'ouvre](#)
- ◆ [Crée un principal kerberos avec les paramètres donnés](#)
- ◆ [Crée un processus de pointeur de fichier](#)
- ◆ [Crée un serveur xmlrpc](#)
- ◆ [Crée un sprite](#)
- ◆ [Crée un tableau](#)
- ◆ [Crée un tableau contenant un intervalle d'éléments](#)
- ◆ [Crée un tableau à partir de deux autres tableaux](#)
- ◆ [Crée un tableau à partir de variables et de leur valeur](#)
- ◆ [Crée un ton pour usage futur](#)
- ◆ [Crée un vecteur d'initialisation à partir d'une source aléatoire](#)
- ◆ [Crée une arbre à partir d'une chaîne xml](#)
- ◆ [Crée une base de données](#)
- ◆ [Crée une base de données dbase](#)
- ◆ [Crée une base de données msql](#)
- ◆ [Crée une base de données mysql](#)
- ◆ [Crée une base lotus notes](#)
- ◆ [Crée une chaîne compressée gzip](#)
- ◆ [Crée une configuration utilisée pour ouvrir un dictionnaire](#)
- ◆ [Crée une connexion persistante](#)
- ◆ [Crée une connexion sur une socket](#)
- ◆ [Crée une copie exacte mais vide d'une relation](#)
- ◆ [Crée une copie vide d'une relation avec les index](#)
- ◆ [Crée une demande de compte ou d'identification](#)
- ◆ [Crée une entité référence](#)
- ◆ [Crée une fonction anonyme \(style lambda\)](#)
- ◆ [Crée une image wbmp](#)
- ◆ [Crée une image à partir d'une chaîne](#)
- ◆ [Crée une instance de type de données win32 et la remplit](#)
- ◆ [Crée une instance pdo qui représente une connexion à une base de données](#)
- ◆ [Crée une note à partir du formulaire form_name](#)

- ◆ Crée une nouvelle action
- ◆ Crée une nouvelle base de données paradox
- ◆ Crée une nouvelle boîte aux lettres
- ◆ Crée une nouvelle brosse
- ◆ Crée une nouvelle fenêtre
- ◆ Crée une nouvelle forme
- ◆ Crée une nouvelle image jpeg à partir d'un fichier ou d'une url
- ◆ Crée une nouvelle image png à partir d'un fichier ou d'une url
- ◆ Crée une nouvelle image wbmp à partir d'un fichier ou d'une url
- ◆ Crée une nouvelle image xbm à partir d'un fichier ou d'une url
- ◆ Crée une nouvelle image xpm à partir d'un fichier ou d'une url
- ◆ Crée une nouvelle image en vraies couleurs
- ◆ Crée une nouvelle image à palette
- ◆ Crée une nouvelle image à partir d'un fichier gd ou d'une url
- ◆ Crée une nouvelle image à partir d'un fichier gd2 ou d'une url
- ◆ Crée une nouvelle image à partir d'un fichier ou d'une url
- ◆ Crée une nouvelle image à partir d'une partie de fichier gd2 ou d'une url
- ◆ Crée une nouvelle instance de la classe hw_api_attribute
- ◆ Crée une nouvelle instance de la classe hw_api_content
- ◆ Crée une nouvelle instance de la classe hw_api_object
- ◆ Crée une nouvelle police
- ◆ Crée une nouvelle relation db++
- ◆ Crée une nouvelle session
- ◆ Crée une nouvelle session muscat
- ◆ Crée une nouvelle transaction à vide
- ◆ Crée une paire de sockets connectées et indissociables
- ◆ Crée une paire de sockets identiques et les stocke dans un tableau
- ◆ Crée une ressource radius pour l'identification
- ◆ Crée une ressource radius pour les comptes
- ◆ Crée une référence sur un composant com (déconseillé)
- ◆ Crée une session
- ◆ Crée une socket
- ◆ Crée une socket serveur unix ou internet
- ◆ Crée une tache de couleur
- ◆ Crée une variable php à partir d'une valeur linéarisée
- ◆ Crée un diff binaire de deux chaînes
- ◆ Crée un diff binaire de deux fichiers
- ◆ Crée un diff unifié entre deux chaînes
- ◆ Crée un diff unifié entre deux fichiers
- ◆ Crée un nouvel itérateur à partir d'une instance arrayobject
- **D**
- - ◆ Demande au serveur de terminer un thread mysql
 - ◆ Demande d'informations d'arbre sur une entité du réseau
 - ◆ Demande des informations sur le serveur interbase
 - ◆ Demande des statistiques sur une base de données interbase
 - ◆ Demande un fichier via scp
 - ◆ Demande un shell interactif
 - ◆ Demande à la bibliothèque de lire un fichier de configuration donné
 - ◆ Demande à readline de refaire l'affichage
 - ◆ Desagrège un objet
 - ◆ Dessine la ligne le long du chemin
 - ◆ Dessine le bord d'une fenêtre avec un caractère qualifié
 - ◆ Dessine le chemin courant
 - ◆ Dessine le premier caractère dans la chaîne de caractères donnée dans la forme en

utilisant la définition de glyphe de la police donnée

- ◆ Dessine un arc dans le sens contraire des aiguilles d'une montre
- ◆ Dessine un arc dans le sens des aiguilles d'une montre
- ◆ Dessine un arc de cercle
- ◆ Dessine un arc de cercle
- ◆ Dessine un arc de cercle
- ◆ Dessine un arc de cercle dans le sens horaire
- ◆ Dessine un arc de cercle pdf dans le sens antihoraire
- ◆ Dessine un arc de radius r centré à la position courante, d'un angle startangle à endangle mesuré dans le sens des aiguilles d'une montre à partir de midi
- ◆ Dessine un bord autour de l'écran avec les caractères données
- ◆ Dessine un caractère horizontalement
- ◆ Dessine un caractère verticalement
- ◆ Dessine un cercle
- ◆ Dessine un cercle
- ◆ Dessine un cercle dans un document pdf
- ◆ Dessine un cercle de radius r centré à la position courante, dans le sens inverse des aiguilles d'une montre
- ◆ Dessine un fichier bmp
- ◆ Dessine un pixel
- ◆ Dessine un polygone
- ◆ Dessine un polygone rempli
- ◆ Dessine un rectangle
- ◆ Dessine un rectangle rempli
- ◆ Dessine un rectangle, avec des coins arrondis
- ◆ Dessine un secteur angulaire
- ◆ Dessine un texte
- ◆ Dessine un texte avec une police truetype
- ◆ Dessine un texte sur une image avec une police postscript type1
- ◆ Dessine une chaîne horizontale
- ◆ Dessine une chaîne verticale
- ◆ Dessine une courbe
- ◆ Dessine une courbe
- ◆ Dessine une courbe
- ◆ Dessine une courbe bézier cubique
- ◆ Dessine une courbe de bezier
- ◆ Dessine une courbe de bézier cubique en utilisant la position courante et les trois points donnés en tant que points de contrôle
- ◆ Dessine une courbe de bézier cubique en utilisant la position courante et les trois points donnés en tant que position de contrôle
- ◆ Dessine une courbe de bézier quadratique entre deux points
- ◆ Dessine une courbe relativement
- ◆ Dessine une ellipse
- ◆ Dessine une ellipse
- ◆ Dessine une ellipse partielle
- ◆ Dessine une ellipse partielle et la remplit
- ◆ Dessine une ellipse pleine
- ◆ Dessine une ligne
- ◆ Dessine une ligne
- ◆ Dessine une ligne

- ◆ Dessine une ligne
- ◆ Dessine une ligne
- ◆ Dessine une ligne
- ◆ Dessine une ligne horizontale à la position courante, en utilisant un caractère et une taille maximale
- ◆ Dessine une ligne horizontale à partir de la position courante, avec un caractère qualifié, et d'une taille maximale
- ◆ Dessine une ligne le long du chemin
- ◆ Dessine une ligne pdf
- ◆ Dessine une ligne pointillée
- ◆ Dessine une ligne relativement
- ◆ Dessine une ligne verticale dans une fenêtre, à la position courante, en utilisant un caractère qualifié, avec une taille maximale
- ◆ Dessine une ligne verticale à la position courante en utilisant un caractère donné, pour une taille donnée
- ◆ Dessine une ligne, relativement
- ◆ Distance de hamming
- ◆ Divise deux nombres de grande taille
- ◆ Divise deux nombres gmp
- ◆ Divise deux variantes et retourne le reste
- ◆ Division exacte de nombres gmp
- ◆ Divisions de 2 nombres gmp
- ◆ Duplique le process courant
- ◆ Duplique un objet blob
- ◆ Décharge le module de chiffrement
- ◆ Déchiffre
- ◆ Déchiffre des données avec une clé privée
- ◆ Déchiffre des données avec une clé publique
- ◆ Déchiffre un message s/mime
- ◆ Déchiffre un texte avec les paramètres donnés
- ◆ Déclenche une erreur utilisateur
- ◆ Décodage en base64 des données pour cybercash
- ◆ Décode des données binaires, passées à une udf sqlite
- ◆ Décode des en-têtes mime multiples
- ◆ Décode le code xml en types php natifs
- ◆ Décode le code xml en une liste de descriptions de méthodes
- ◆ Décode le code xml en variables php natives
- ◆ Décode les données de session
- ◆ Décode les entités html en caractères
- ◆ Décode les éléments mime d'un en-tête
- ◆ Décode un champ d'en-tête mime
- ◆ Décode un en-tête mime
- ◆ Décode un pseudo encodé
- ◆ Décode un texte encodé en base64
- ◆ Décode une chaîne au format uuencode
- ◆ Décode une chaîne en mime base64
- ◆ Décode une chaîne encodée avec addcslashes
- ◆ Décode une chaîne encodée en utf-7 modifié
- ◆ Décode une chaîne encodée url
- ◆ Décode une chaîne url
- ◆ Décode une liste de paramètres js-encodés
- ◆ Décompresse une chaîne
- ◆ Décompresse une chaîne bzip2
- ◆ Décompresse une chaîne compressée

- ◆ Décompression lzf
- ◆ Déconditionne des données depuis une chaîne binaire
- ◆ Déconfigure le mode brut du terminal
- ◆ Déconnecte d'un serveur ldap
- ◆ Déconnecte un serveur maxdb
- ◆ Déconnexion d'une base sesam
- ◆ Déconnexion du service de gestion interbase
- ◆ Décrit la bibliothèque dbm utilisée
- ◆ Décrit la transition utilisée pour déclencher une liste d'actions
- ◆ Décryptage cybercash
- ◆ Décrémente la valeur d'un élément
- ◆ Décrémente le compteur de références
- ◆ Dédoublonne un tableau
- ◆ Définit un code de clé (keycode)
- ◆ Définit \$newval comme valeur à l'\$index spécifié
- ◆ Définit des options
- ◆ Définit l'index
- ◆ Définit l'index par un nombre
- ◆ Définit l'objet dans lequel doivent être résolues les fonctions de callback
- ◆ Définit la valeur d'un attribut
- ◆ Définit la valeur d'un paramètre
- ◆ Définit un attribut
- ◆ Définit un attribut de requête
- ◆ Définit un attribut étendu
- ◆ Définit une propriété d'auditeur
- ◆ Définit une propriété de la source
- ◆ Définit l'encodage pour les caractères des champs (déprécié)
- ◆ Définit la classe qui gère les requêtes soap
- ◆ Définit la couleur 0
- ◆ Définit la couleur transparente
- ◆ Définit le cookie qui sera envoyé avec la requête soap
- ◆ Définit le décalage horaire par défaut utilisé par toutes les fonctions date/heure dans un script
- ◆ Définit le fichier depuis lequel les blobs seront lus
- ◆ Définit le fond d'écran
- ◆ Définit le jeu de caractères par défaut du client
- ◆ Définit le mode de récupération par défaut pour cette requête
- ◆ Définit le mode persistant de soapserver
- ◆ Définit le nom de la table (déprécié)
- ◆ Définit le point de vue de l'utilisateur en coordonnées polaires
- ◆ Définit le pointeur de résultat à une position donnée
- ◆ Définit les attributs donnés à une étiquette fonction-key
- ◆ Définit les données de session
- ◆ Définit les options
- ◆ Définit les options d'un processeur xsl donné
- ◆ Définit les propriétés de fond d'écran pour le terminal
- ◆ Définit les variables msession depuis un tableau
- ◆ Définit lines pour inisr() et newterm() à 1
- ◆ Définit un paramètre
- ◆ Définit un polygone flash
- ◆ Définit un rectangle
- ◆ Définit un symbole
- ◆ Définit un type pour utilisation avec les fonctions windows api
- ◆ Définit une chaîne de texte

- ◆ Définit une constante
- ◆ Définit une couche mask à un niveau
- ◆ Définit une fonction utilisateur de gestion d'exceptions
- ◆ Définit une image bitmap
- ◆ Définit une ligne
- ◆ Définit une police flash
- ◆ Définit une projection orthogonale entre les coordonnées utilisateur et le port courant
- ◆ Définit une projection orthogonale à 2 dimensions entre les coordonnées utilisateur et le port courant
- ◆ Définit une projection orthogonale à 3 dimensions entre les coordonnées utilisateur et le port courant
- ◆ Définit une transformation de vue
- ◆ Définit une valeur
- ◆ Délinéarise un paquet wddx
- ◆ Démarre la lecture de la source
- ◆ Démarre la session de débogage
- ◆ Démarre la session de débogage à distance
- ◆ Démarre la session de déboguage apd
- ◆ Démarre le serveur web virtuel
- ◆ Démarre le service iis identifié par serviceid
- ◆ Démarre le streaming d'une source
- ◆ Démarre tls
- ◆ Démarre un nouveau modèle
- ◆ Démarre un nouveau motif
- ◆ Démarre une base de données fbsql
- ◆ Démarre une nouvelle page
- ◆ Démarre une nouvelle transaction
- ◆ Démarre une section de texte
- ◆ Démarre une transaction
- ◆ Dépasse les fonctions intégrées
- ◆ Dépile la matrice de transformation
- ◆ Dépile un élément au début d'un tableau
- ◆ Dépile un élément de la fin d'un tableau
- ◆ Déplace des messages dans une boîte aux lettres
- ◆ Déplace immédiatement le curseur
- ◆ Déplace l'itérateur à la position suivante
- ◆ Déplace l'origine de l'objet swffill
- ◆ Déplace la position courante et ajoute un caractère
- ◆ Déplace la position d'affichage
- ◆ Déplace la position d'affichage de la fenêtre
- ◆ Déplace la position et ajoute la chaîne attribuée avec la taille donnée
- ◆ Déplace la position et ajoute une chaîne
- ◆ Déplace la position et ajoute une chaîne attribuée
- ◆ Déplace la position et ajoute une chaîne de taille donnée
- ◆ Déplace la position et efface le caractère, puis décale le reste de la ligne à gauche
- ◆ Déplace la position et lit le caractère à la nouvelle position
- ◆ Déplace la position et lit le caractère à la nouvelle position
- ◆ Déplace le pointeur de lecture
- ◆ Déplace le pointeur de résultat
- ◆ Déplace le pointeur de résultat mysql

- ◆ Déplace le pointeur de résultat sqlite vers une ligne
- ◆ Déplace le pointeur de résultat sur le champ spécifié
- ◆ Déplace le pointeur de résultat vers une position donnée
- ◆ Déplace le pointeur interne d'un lob oracle
- ◆ Déplace le pointeur interne de fichier
- ◆ Déplace le pointeur interne de ligne
- ◆ Déplace le pointeur interne de lignes sybase
- ◆ Déplace le pointeur interne de résultat
- ◆ Déplace le pointeur interne de résultat fbsql
- ◆ Déplace le pointeur interne de résultat mysql
- ◆ Déplace le pointeur interne ms sql server au résultat suivant
- ◆ Déplace le pointeur interne msql
- ◆ Déplace le pointeur interne vers le résultat suivant
- ◆ Déplace le pointeur sqlite vers la prochaine ligne
- ◆ Déplace le stylo
- ◆ Déplace le stylo de texte
- ◆ Déplace le stylo relativement
- ◆ Déplace un conteneur pour que son coin supérieur gauche soit aux coordonnées [startx, starty]
- ◆ Déplace un curseur à défilement
- ◆ Déplace un fichier téléchargé
- ◆ Déplace un objet
- ◆ Déplace un objet d'une collection à l'autre
- ◆ Déplace un objet en coordonnées globales
- ◆ Déplace un objet en coordonnées relatives
- ◆ Dérive les clés mppe depuis des données
- ◆ Désactive l'analyseur rpl
- ◆ Désactive l'attribut donné pour les étiquettes des fonction-key (fonctions clés)
- ◆ Désactive l'écho clavier
- ◆ Désactive la lecture depuis le maître
- ◆ Désactive la lecture depuis le maître
- ◆ Désactive la validation automatique oracle
- ◆ Désactive le déboguer interne de php (php 3)
- ◆ Désactive le parseur rpl
- ◆ Désactive le rapport d'erreur libxml et les stocke pour lecture ultérieure
- ◆ Désactive le remplissage de formes flash
- ◆ Désactive les attributs d'une fenêtre
- ◆ Désactive les attributs donnés
- ◆ Détecte l'encodage mime
- ◆ Détecte le type d'encodage d'un caractère http
- ◆ Détecte le version du code html utilisée dans un document
- ◆ Détecte un encodage
- ◆ Détermine l'accessibilité d'un fichier
- ◆ Détermine la le degré des messages retournés par pg_last_error et pg_result_error
- ◆ Détermine le meilleur encodage pour un fichier
- ◆ Détermine le mode d'optimisation de l'extension lzf
- ◆ Détermine le nombre de décimales par défaut
- ◆ Détermine le rendu du texte
- ◆ Détermine le type d'une image
- ◆ Détermine si l'argument peut être appelé comme fonction
- ◆ Détermine si la valeur de retour des fonctions courantes sera utilisée
- ◆ Détermine si un objet est une sous-classe
- ◆ Détermine si un pointeur de fichier est un terminal interactif
- ◆ Détermine si une extension est chargée ou non

- ◆ Détermine si une image est une image truecolor
- ◆ Détermine si une valeur de tableau représente un xmlrpc
- ◆ Détermine si une variable contient une valeur non nulle
- ◆ Détermine si une variable est affectée
- ◆ Détermine si une variable est de type chaîne de caractères
- ◆ Détermine si une variable est de type nombre décimal
- ◆ Détermine si une variable est de type nombre entier
- ◆ Détermine si une variable est de type objet
- ◆ Détermine si une variable est un booléen
- ◆ Détermine si une variable est un tableau
- ◆ Détermine si une variable est un type numérique
- ◆ Détermine si une variable est une ressource
- ◆ Détruit la connexion et la structure mcve_conn
- ◆ Détruit la session
- ◆ Détruit les données du buffer de sortie et éteint la bufferisation de sortie
- ◆ Détruit toutes les variables d'une session
- ◆ Détruit un analyseur xslt
- ◆ Détruit un analyseur créé par mailparse_msg_create
- ◆ Détruit un analyseur xml
- ◆ Détruit un bloc de mémoire partagée
- ◆ Détruit un buffer openal
- ◆ Détruit un contexte
- ◆ Détruit un document
- ◆ Détruit un pointeur de lob oracle
- ◆ Détruit un serveur xmlrpc
- ◆ Détruit un sémaphore
- ◆ Détruit une file de messages
- ◆ Détruit une image
- ◆ Détruit une relation
- ◆ Détruit une ressource de source
- ◆ Détruit une session
- ◆ Détruit une session mnogosearch
- ◆ Détruit une variable
- ◆ Déverrouille un objet
- ◆ Déverrouille un objet hyperwave verrouillé
- ◆ Déverrouille une session
- **E**
- - ◆ Echappe les caractères spéciaux d'une chaîne afin de l'utiliser dans une requête sql, en tenant compte du jeu de caractères courant de la connexion
 - ◆ Echappe les méta-caractères
 - ◆ Echappement des caractères spéciaux des expressions rationnelles
 - ◆ Eclate une chaîne par expression rationnelle
 - ◆ Ecrase une partie d'un lob oracle
 - ◆ Ecrire dans un bloc de mémoire partagée
 - ◆ Ecrit dans l'historique
 - ◆ Ecrit dans le socket de débogage
 - ◆ Ecrit dans un fichier compressé gzip
 - ◆ Ecrit dans une socket
 - ◆ Ecrit des données dans un lob oracle
 - ◆ Ecrit des données sur l'imprimante
 - ◆ Ecrit du texte dans une image avec la police courante freetype 2
 - ◆ Ecrit l'en-tête bcompiler
 - ◆ Ecrit la valeur courante de l'option quick_print de la bibliothèque ucd
 - ◆ Ecrit le caractère \x00 pour indiquer la fin des données compilées

- ◆ Écrit les données de session et ferme la session
- ◆ Écrit les informations de débogage dans le log
- ◆ Écrit les lob oracle sur le disque
- ◆ Écrit toutes les fonctions définies dans un fichier sous forme de bytecode
- ◆ Écrit toutes les modifications apportées à une relation
- ◆ Écrit un code source php sous forme de bytecodes
- ◆ Écrit un document pdf dans un fichier
- ◆ Écrit un fichier en mode binaire
- ◆ Écrit un fichier sur un serveur ftp, et le lit depuis un fichier (non bloquant)
- ◆ Écrit un lob oracle temporaire
- ◆ Écrit un objet de grande taille postgresql
- ◆ Écrit une chaîne dans un fichier
- ◆ Écrit une chaîne dans un objet slob
- ◆ Écrit une chaîne formatée dans un flux
- ◆ Écrit une chaîne formatée dans un flux
- ◆ Écrit une classe définie en bytecode
- ◆ Écrit une constante définie comme bytecode
- ◆ Écrit une fonction définie sous forme de bytecode
- ◆ Écrit à la position start et jusqu'à la fin du fichier de type exe
- ◆ Écriture binaire dans un fichier bzip2
- ◆ Efface des lignes postgresql
- ◆ Efface des objets
- ◆ Efface dynamiquement une méthode donnée
- ◆ Efface et remplace une portion de tableau
- ◆ Efface l'historique
- ◆ Efface l'écran
- ◆ Efface l'écran depuis la position courante jusqu'au bas de l'écran
- ◆ Efface l'écran depuis la position courante jusqu'à la fin de la ligne
- ◆ Efface l'écran du terminal
- ◆ Efface la dernière exception java
- ◆ Efface la fenêtre
- ◆ Efface la file d'attente de l'imprimante
- ◆ Efface la ligne courante, et monte l'écran d'une ligne
- ◆ Efface la ressource d'une base de données paradox
- ◆ Efface la valeur à l'\$index spécifié
- ◆ Efface le buffer de sortie
- ◆ Efface le cache de stat
- ◆ Efface le caractère courant et décale le reste de la ligne vers la gauche
- ◆ Efface le contenu de la fenêtre
- ◆ Efface les attributs de la fenêtre
- ◆ Efface les soft labels de l'écran
- ◆ Efface tous les alias d'un utilisateur
- ◆ Efface tous les alias virtuels d'un domaine
- ◆ Efface tous les messages marqués pour l'effacement
- ◆ Efface tous les éléments existant sur le serveur de cache
- ◆ Efface toutes les erreurs précédemment générées par une socket
- ◆ Efface un attribut
- ◆ Efface un attribut
- ◆ Efface un attribut
- ◆ Efface un attribut à l'entrée courante
- ◆ Efface un attribut étendu
- ◆ Efface un calendrier
- ◆ Efface un contexte d'imprimante
- ◆ Efface un domaine virtuel

- ◆ [Efface un domaine virtuel](#)
- ◆ [Efface un dossier](#)
- ◆ [Efface un dossier](#)
- ◆ [Efface un dossier ftp](#)
- ◆ [Efface un enregistrement dans une base dbase](#)
- ◆ [Efface un fichier](#)
- ◆ [Efface un fichier](#)
- ◆ [Efface un fichier sur un serveur ftp](#)
- ◆ [Efface un gestionnaire de callback installé précédemment et restaure les paramètres du terminal](#)
- ◆ [Efface un noeud](#)
- ◆ [Efface un objet](#)
- ◆ [Efface un objet de grande taille postgresql](#)
- ◆ [Efface un objet pdf](#)
- ◆ [Efface un paramètre](#)
- ◆ [Efface un principal kerberos](#)
- ◆ [Efface un stylo](#)
- ◆ [Efface un tag id3](#)
- ◆ [Efface un utilisateur d'un domaine virtuel](#)
- ◆ [Efface un utilisateur d'une base de données de sécurité \(uniquement pour ib6 ou plus récent\)](#)
- ◆ [Efface un élément du serveur de cache](#)
- ◆ [Efface un événement dans un agenda icap](#)
- ◆ [Efface un événement dans un calendrier mcal](#)
- ◆ [Efface une base de données msql](#)
- ◆ [Efface une base de données mysql](#)
- ◆ [Efface une boîte aux lettres](#)
- ◆ [Efface une brosse](#)
- ◆ [Efface une entrée dans un dossier](#)
- ◆ [Efface une fenêtre ncurses](#)
- ◆ [Efface une ligne dans une base dba](#)
- ◆ [Efface une police](#)
- ◆ [Efface une valeur](#)
- ◆ [Efface une variable de la mémoire partagée](#)
- ◆ [Effacer](#)
- ◆ [Effectue une négation logique sur un variant](#)
- ◆ [Effectue des actions de débogage](#)
- ◆ [Effectue des opérations de débogage](#)
- ◆ [Effectue la césure d'une chaîne](#)
- ◆ [Effectue les opérations de nettoyage et de réparation préparées pour un fichier html](#)
- ◆ [Effectue un et entre deux variants et retourne le résultat](#)
- ◆ [Effectue un test d'autorisation de crédit sur une transaction](#)
- ◆ [Effectue une communication avec mcve \(envoi/réception de données\) en mode non-bloquant](#)
- ◆ [Effectue une recherche](#)
- ◆ [Effectue une requête partielle pour l'uri spécifiée et renvoie toutes les informations](#)
- ◆ [Effectue une requête soap](#)
- ◆ [Effectue une rotation](#)
- ◆ [Effectue une sauvegarde de base de données interbase](#)
- ◆ [Effectue une sous-requête apache](#)
- ◆ [Effectue une sous-requête nsapi](#)
- ◆ [Effectue une sélection postgresql](#)
- ◆ [Effectue une transaction avec payflow pro](#)
- ◆ [Effectue une transformation rot13](#)

- ◆ Effectue une translation de l'origine du système de coordonnées
- ◆ Effectue une validation relaxng sur le document
- ◆ Effectue une validation relaxng sur le document
- ◆ Effectue une vérification de mot de passe
- ◆ Elève les privilèges au rang d'opérateur irc
- ◆ Elève un nombre à la puissance n-ième
- ◆ Emet les en-têtes http spécifiques à fdf
- ◆ Empile la matrice de transformations courante dans la pile
- ◆ Empile un ou plusieurs éléments au début d'un tableau
- ◆ Empile un ou plusieurs éléments à la fin d'un tableau
- ◆ Enclenche la bufferisation de sortie
- ◆ Encodage en base64 pour cybercash
- ◆ Encode des entités html
- ◆ Encode les caractères spéciaux dans un pseudo pour être compatible avec irc
- ◆ Encode les données binaires d'une udf sqlite avant de les retourner
- ◆ Encode les données de session
- ◆ Encode une chaîne de caractères en utilisant l'algorithme uuencode
- ◆ Encode une chaîne en mime base64
- ◆ Encode une chaîne en url
- ◆ Encode une chaîne en url, selon la rfc 1738
- ◆ Encode une chaîne pour un en-tête mime
- ◆ Encryptage cybercash
- ◆ Endroit où se trouve le fichier global des mots
- ◆ Endroit où se trouvent les fichiers de données linguistiques
- ◆ Enlève la marque d'effacement d'un message
- ◆ Enlève un objet
- ◆ Enlève une définition de fonction
- ◆ Enlève/supprime une constante déjà définie
- ◆ Enregistre l'espace de noms avec l'objet domxpath
- ◆ Enregistre plusieurs valeurs dans un paquet wddx
- ◆ Enregistre un filtre de flux
- ◆ Enregistre un nouveau format d'affichage des messages irc
- ◆ Enregistre un nouvel événement dans un calendrier mcal
- ◆ Enregistre un événement dans un agenda icap
- ◆ Enregistre une enveloppe url, implémentée comme une classe php
- ◆ Enregistre une fonction de callback sur un événement ibase
- ◆ Enregistre une fonction de complétion
- ◆ Enregistre une fonction exécutée à chaque tick
- ◆ Enregistre une fonction php avec une méthode
- ◆ Enregistre une fonction php pour générer la documentation
- ◆ Enregistre une fonction pour exécution à l'extinction
- ◆ Enregistre une fonction utilisateur "classique" udf pour sqlite
- ◆ Enregistre une fonction win32 dans php
- ◆ Enregistre une paire de remplacement pour un mot
- ◆ Enregistre une udf agregeante pour les requêtes sqlite
- ◆ Enregistre une valeur dans un paquet wddx
- ◆ Enregistre une variable dans une session
- ◆ Envoi des données dans un block
- ◆ Envoi un mail
- ◆ Envoi une requête et retourne
- ◆ Envoi une requête à un serveur mysql
- ◆ Envoie des données mysql par paquets
- ◆ Envoie des données à une socket connectée
- ◆ Envoie la commande site au serveur

- ◆ Envoie la demande et attente une réponse
- ◆ Envoie la requête et retourne
- ◆ Envoie le buffer de sortie
- ◆ Envoie les données du buffer de sortie et éteint la bufferisation de sortie
- ◆ Envoie tout le contenu généré dans un fichier
- ◆ Envoie un cookie
- ◆ Envoie un cookie sans encoder sa valeur en url
- ◆ Envoie un en-tête http
- ◆ Envoie un fichier mp3 en streaming
- ◆ Envoie un fichier sur le serveur ftp (non-bloquant)
- ◆ Envoie un fichier via scp
- ◆ Envoie un mail encodé
- ◆ Envoie un message dans une file
- ◆ Envoie un message mail
- ◆ Envoie un message à un canal ou un utilisateur
- ◆ Envoie un signal à un processus
- ◆ Envoie une chaîne au serveur postgresql
- ◆ Envoie une chaîne à l'api de base muscat
- ◆ Envoie une commande au serveur et attend le résultat, avec les capacités de passer des paramètres séparément de la commande texte sql
- ◆ Envoie une commande et sépare les paramètres au serveur sans attendre le(s) résultat(s)
- ◆ Envoie une commande ftp brute
- ◆ Envoie une image gif vers un navigateur ou un fichier
- ◆ Envoie une image jpeg vers un navigateur ou un fichier
- ◆ Envoie une image png vers un navigateur ou un fichier
- ◆ Envoie une message à la socket, connectée ou pas
- ◆ Envoie une message à une socket, qu'elle soit connectée ou pas
- ◆ Envoie une note (notice) à un utilisateur
- ◆ Envoie une requête et lit tous les résultats dbx
- ◆ Envoie une requête informix
- ◆ Envoie une requête msql
- ◆ Envoie une requête mysql à un serveur mysql
- ◆ Envoie une requête pour créer une requête préparée avec les paramètres donnés et attend l'exécution
- ◆ Envoie une requête pour créer une requête préparée avec les paramètres donnés, sans attendre la fin de son exécution
- ◆ Envoie une requête pour exécuter une requête préparée avec des paramètres donnés, sans attendre le(s) résultat(s)
- ◆ Envoie une requête pour exécuter une requête préparée avec les paramètres donnés et attend le résultat
- ◆ Envoie une requête sql au serveur ms sql
- ◆ Envoie une requête sql à un serveur ingres ii
- ◆ Envoie une requête à la base frontbase
- ◆ Envoie une requête à sybase et ne bloque pas
- ◆ Envoie une requête à un serveur cyrus imap
- ◆ Envoie une requête à une base sybase
- ◆ Envoie une transaction brute à payflow pro
- ◆ Envoie votre animation au navigateur
- ◆ Et logique
- ◆ Etablit le diagnostic pour le document analysé et réparé
- ◆ Etablit une connexion avec un serveur oracle
- ◆ Etablit une nouvelle connexion à oracle
- ◆ Eteind l'alarme d'un événement

- ◆ Eteind l'alarme d'un événement
- ◆ Eteint le serveur embarqué
- ◆ Eteint un serveur embarqué
- ◆ Eteint une socket en lecture et/ou écriture
- ◆ Etend ou condense une police de caractères
- ◆ Etire un objet en coordonnées globales
- ◆ Etire un objet relativement
- ◆ Evalue l'expression xpath donnée
- ◆ Evalue une expression xpath donnée et retourne un résultat écrit si possible
- ◆ Exemples d'utilisation du gestionnaire ogg://.
- ◆ Exponentielle
- ◆ Exporte un certificat vers un fichier
- ◆ Exporte un certificat vers une variable
- ◆ Exporte un csr vers un fichier ou une variable
- ◆ Exporte un lob oracle dans un fichier
- ◆ Exporte un objet de grande taille vers un fichier
- ◆ Exporte une csr vers un fichier
- ◆ Expression exponentielle
- ◆ Expression rationnelle avec support des caractères multi-octets, sans tenir compte de la casse
- ◆ Expression rationnelle globale
- ◆ Expression rationnelle insensible à la casse avec le support des caractères multi-octets
- ◆ Expression rationnelle posix pour les chaînes multi-octets
- ◆ Expression rationnelle standard
- ◆ Expression rationnelle standard
- ◆ Expulse un utilisateur d'un canal
- ◆ Extrait des informations d'un résultat
- ◆ Extrait et décode une section de message
- ◆ Extrait et décode une section de message
- ◆ Extrait les informations d'une référence d'entrée
- ◆ Extrait toutes les balises méta d'un fichier html
- ◆ Extrait un attribut
- ◆ Extrait un attribut spécifique au vendeur
- ◆ Extrait un fichier intégré dans un document fdf
- ◆ Extrait un morceaux de données dans le noeud
- ◆ Extrait un objet
- ◆ Extrait un élément d'une archive rar
- ◆ Extrait une clé privée d'un certificat, et la prépare
- ◆ Extrait une portion de tableau
- ◆ Extrait une représentation sous forme de chaîne d'une valeur interne à zend pour affichage
- ◆ Exécute du code tcl côté serveur
- ◆ Exécute immédiatement une requête sql
- ◆ Exécute l'appel système select() un tableau de sockets avec une durée d'expiration
- ◆ Exécute le programme indiqué dans l'espace courant de processus
- ◆ Exécute les sous-requêtes de la session curl
- ◆ Exécute un programme externe
- ◆ Exécute un programme externe et affiche le résultat
- ◆ Exécute un programme externe et affiche le résultat brut
- ◆ Exécute une chaîne comme un script php
- ◆ Exécute une commande analysée sur un pointeur oracle
- ◆ Exécute une commande de maintenance sur une base de données interbase
- ◆ Exécute une commande et ouvre les pointeurs de fichiers pour les entrées / sorties

- ◆ Exécute une commande spécifique à un protocole et qui n'est donc pas disponible dans l'api générique
 - ◆ Exécute une commande sql oracle
 - ◆ Exécute une commande sur le serveur ftp
 - ◆ Exécute une commande sur un serveur distant
 - ◆ Exécute une commande via le shell et retourne le résultat sous forme de chaîne
 - ◆ Exécute une fonction fichier de langage c
 - ◆ Exécute une fonction sur chacun des éléments d'un tableau
 - ◆ Exécute une procédure stockée sur un serveur ms sql
 - ◆ Exécute une requête aql
 - ◆ Exécute une requête et retourne soit un tableau pour une colonne unique, soit la valeur de la première ligne
 - ◆ Exécute une requête ibase préparée
 - ◆ Exécute une requête mysql multiple
 - ◆ Exécute une requête ovrimos préparée
 - ◆ Exécute une requête postgresql
 - ◆ Exécute une requête postgresql asynchrone
 - ◆ Exécute une requête préparée
 - ◆ Exécute une requête préparée
 - ◆ Exécute une requête préparée
 - ◆ Exécute une requête sans résultats sur une base de données
 - ◆ Exécute une requête sesam
 - ◆ Exécute une requête sql
 - ◆ Exécute une requête sql avec sqlite et retourne un tableau
 - ◆ Exécute une requête sql et retourne le nombre de lignes affectées
 - ◆ Exécute une requête sql préparée
 - ◆ Exécute une requête sql préparée
 - ◆ Exécute une requête sql sans mobiliser les résultats mysql
 - ◆ Exécute une requête sql, retourne un jeu de résultats en tant qu'objet pdostatement
 - ◆ Exécute une requête sqlite et lit le résultat
 - ◆ Exécute une requête sqlite non bufférisée
 - ◆ Exécute une requête sur la base de données
 - ◆ Exécute une requête sur un serveur frontbase
 - ◆ Exécute une requête sur une base de données
 - ◆ Exécute une requête sur une base de données
 - ◆ Exécute une requête sur une base ibase
 - ◆ Exécute une requête xpath sur des données xml
 - ◆ Exécute une session curl
- F
- ◆ Factorielle gmp
 - ◆ Fait beeper le terminal
 - ◆ Fait correspondre un peu plus les couleurs de la version palette d'une image aux couleurs de sa version truecolor
 - ◆ Fait du processus courant un chef de session
 - ◆ Fait flasher le terminal (visual bell)
 - ◆ Fait tourner une forme relativement
 - ◆ Fait tourner une image d'un angle
 - ◆ Fait une copie d'une police chargée pour modification ultérieure
 - ◆ Fait une liste détaillée des fichiers dans un dossier
 - ◆ Fait une pause
 - ◆ Fausse documentation pour main

- ◆ [Ferme et trace un chemin](#)
- ◆ [Ferme l'accès au fichier](#)
- ◆ [Ferme l'image et libère la mémoire](#)
- ◆ [Ferme la connexion au serveur msession](#)
- ◆ [Ferme la connexion avec le serveur memcache](#)
- ◆ [Ferme la connexion avec le serveur d'administration et libère toutes les ressources associées](#)
- ◆ [Ferme la connexion avec un serveur](#)
- ◆ [Ferme la connexion fam](#)
- ◆ [Ferme la connexion frontbase](#)
- ◆ [Ferme la connexion hyperwave](#)
- ◆ [Ferme la connexion mnogosearch avec le document enregistré](#)
- ◆ [Ferme la connexion mysql](#)
- ◆ [Ferme la connexion à l'historique système](#)
- ◆ [Ferme la connexion à muscat](#)
- ◆ [Ferme la connexion à sqlite](#)
- ◆ [Ferme la connexion à un serveur cyrus imap](#)
- ◆ [Ferme la page active](#)
- ◆ [Ferme la page pdf](#)
- ◆ [Ferme le chemin](#)
- ◆ [Ferme le curseur, permettant à la requête d'être de nouveau exécutée](#)
- ◆ [Ferme le document](#)
- ◆ [Ferme le document postscript](#)
- ◆ [Ferme le fichier courant shockwave flash](#)
- ◆ [Ferme le fichier et dessine une ligne le long du chemin](#)
- ◆ [Ferme le fichier pdf](#)
- ◆ [Ferme le fichier pdf d'entrée](#)
- ◆ [Ferme le flux icap](#)
- ◆ [Ferme le pointeur sur le dossier](#)
- ◆ [Ferme toutes les connexions odbc](#)
- ◆ [Ferme un blob ibase](#)
- ◆ [Ferme un bloc de mémoire partagée](#)
- ◆ [Ferme un chemin](#)
- ◆ [Ferme un document fdf](#)
- ◆ [Ferme un dossier d'archive](#)
- ◆ [Ferme un fichier](#)
- ◆ [Ferme un fichier bzip2](#)
- ◆ [Ferme un fichier pdf](#)
- ◆ [Ferme un lob oracle](#)
- ◆ [Ferme un objet de grande taille postgresql](#)
- ◆ [Ferme un objet slob](#)
- ◆ [Ferme un pointeur oracle](#)
- ◆ [Ferme un pointeur sur fichier gz ouvert](#)
- ◆ [Ferme un processus de pointeur de fichier](#)
- ◆ [Ferme un processus ouvert par proc_open et retourne le code de sortie](#)
- ◆ [Ferme un périphérique openal](#)
- ◆ [Ferme une archive zip](#)
- ◆ [Ferme une base dba](#)
- ◆ [Ferme une base dbase](#)
- ◆ [Ferme une base de données dbm](#)
- ◆ [Ferme une base de données paradox](#)
- ◆ [Ferme une connexion](#)
- ◆ [Ferme une connexion ftp](#)
- ◆ [Ferme une connexion mcal](#)

- ◆ [Ferme une connexion ms sql server](#)
- ◆ [Ferme une connexion msql](#)
- ◆ [Ferme une connexion odbc](#)
- ◆ [Ferme une connexion oracle](#)
- ◆ [Ferme une connexion oracle](#)
- ◆ [Ferme une connexion ovrimos](#)
- ◆ [Ferme une connexion sybase](#)
- ◆ [Ferme une connexion yaz](#)
- ◆ [Ferme une connexion à un serveur informix](#)
- ◆ [Ferme une connexion à un serveur ingres](#)
- ◆ [Ferme une connexion à une base dbx](#)
- ◆ [Ferme une connexion à une base de données](#)
- ◆ [Ferme une connexion à une base de données interbase](#)
- ◆ [Ferme une connexion à une imprimante](#)
- ◆ [Ferme une image dans un document pdf](#)
- ◆ [Ferme une relation](#)
- ◆ [Ferme une requête préparée](#)
- ◆ [Ferme une requête préparée](#)
- ◆ [Ferme une session curl](#)
- ◆ [Ferme une socket](#)
- ◆ [Filtre les éléments d'un tableau](#)
- ◆ [Finalise et envoie la transaction](#)
- ◆ [Fixe certaines valeurs](#)
- ◆ [Fixe certains paramètres](#)
- ◆ [Fixe comment les lignes connectés sont jointes](#)
- ◆ [Fixe et lit différentes options d'assertions](#)
- ◆ [Fixe l'alarme de la structure globale](#)
- ◆ [Fixe l'animation de la transition entre les pages](#)
- ◆ [Fixe l'apparence d'un champ fdf](#)
- ◆ [Fixe l'apparence d'une ligne pointillée](#)
- ◆ [Fixe l'apparence d'une ligne pointillée](#)
- ◆ [Fixe l'apparence des fins de ligne](#)
- ◆ [Fixe l'espacement des caractères](#)
- ◆ [Fixe l'espacement des mots](#)
- ◆ [Fixe l'identifiant de groupe de processus](#)
- ◆ [Fixe l'offset d'un champ msql](#)
- ◆ [Fixe l'offset du pointeur de champ ms sql server](#)
- ◆ [Fixe l'uid effective du processus courant](#)
- ◆ [Fixe l'échelle horizontale du texte](#)
- ◆ [Fixe l'élévation du texte](#)
- ◆ [Fixe la catégorie de la structure globale](#)
- ◆ [Fixe la classe de la structure globale](#)
- ◆ [Fixe la couleur courante](#)
- ◆ [Fixe la couleur des bordures pour les commentaires](#)
- ◆ [Fixe la couleur globale d'addition flash \(? : the global add color\)](#)
- ◆ [Fixe la couleur globale de multiplication \(? : the global multiply color\)](#)
- ◆ [Fixe la couleur pour le style courant de remplissage flash](#)
- ◆ [Fixe la date de début de la structure globale](#)
- ◆ [Fixe la date de fin de la structure globale](#)
- ◆ [Fixe la description de la structure globale](#)
- ◆ [Fixe la distance entre deux lignes](#)
- ◆ [Fixe la grandeur des tirets pour les bordures des commentaires](#)
- ◆ [Fixe la largeur d'une ligne](#)
- ◆ [Fixe la largeur de ligne](#)

- ◆ Fixe la matrice de transformation de l'item
- ◆ Fixe la matrice du texte
- ◆ Fixe la méthode de connexion à drop-file
- ◆ Fixe la méthode de connexion à ip
- ◆ Fixe la méthode de connexion à ssl
- ◆ Fixe la page courante
- ◆ Fixe la platitude (flatness)
- ◆ Fixe la police à utiliser pour la prochaine écriture
- ◆ Fixe la position du texte
- ◆ Fixe la position l'écriture du texte
- ◆ Fixe la position à plat
- ◆ Fixe la récurrence
- ◆ Fixe la récurrence annuelle
- ◆ Fixe la récurrence hebdomadaire
- ◆ Fixe la récurrence mensuelle
- ◆ Fixe la récurrence quotidienne
- ◆ Fixe la sévérité minimale du client pour le serveur sybase
- ◆ Fixe la sévérité minimale du client pour les erreurs
- ◆ Fixe la sévérité minimale du client pour les messages
- ◆ Fixe la sévérité minimale du client sybase
- ◆ Fixe la valeur d'une variable d'environnement
- ◆ Fixe la valeur de gris
- ◆ Fixe la valeur de la clé /status
- ◆ Fixe le cadre courant
- ◆ Fixe le chemin d'un domaine
- ◆ Fixe le contexte des actions
- ◆ Fixe le créateur d'un document pdf
- ◆ Fixe le domaine par défaut
- ◆ Fixe le facteur de redimensionnement
- ◆ Fixe le facteur de rotation
- ◆ Fixe le fichier courant ou la position courante
- ◆ Fixe le fichier ssl ca (certificate authority) pour vérification du certificat du serveur
- ◆ Fixe le format de date pour les prochaines requêtes
- ◆ Fixe le gid effectif du processus courant
- ◆ Fixe le mode bloquant/non-bloquant pour une connexion
- ◆ Fixe le motif de pointillés
- ◆ Fixe le niveau de rapport d'erreurs php
- ◆ Fixe le niveau de seuil cubique (??)
- ◆ Fixe le niveau de sévérité des erreurs ms sql server
- ◆ Fixe le niveau de sévérité des messages d'erreurs ms sql server
- ◆ Fixe le nombre maximum d'images dans le sprite
- ◆ Fixe le paramètre linecap
- ◆ Fixe le paramètre linejoin
- ◆ Fixe le paramètre miter limit
- ◆ Fixe le point courant
- ◆ Fixe le point courant de dessin cpdf
- ◆ Fixe le point courant relativement
- ◆ Fixe le remplissage d'un champ de texte
- ◆ Fixe le style courant de ligne
- ◆ Fixe le style de bordure des commentaires
- ◆ Fixe le sujet d'un document pdf
- ◆ Fixe le temps maximal d'une transaction (par transaction)
- ◆ Fixe le temps maximum d'exécution d'un script
- ◆ Fixe le titre d'un document pdf

- ◆ [Fixe le titre de la structure globale](#)
- ◆ [Fixe les champs d'informations d'un document](#)
- ◆ [Fixe les fichiers de clé de certificat et certifie si le serveur requiert la vérification du client par certificat](#)
- ◆ [Fixe les limites d'un document pdf](#)
- ◆ [Fixe les limites de l'anglet](#)
- ◆ [Fixe les mots-clés d'un document pdf](#)
- ◆ [Fixe une translation](#)
- ◆ [Fonction de callback ob_start pour réparer le buffer](#)
- ◆ [Fonction de callback pour la compression automatique des buffers](#)
- ◆ [Fonction de traitement des affichages](#)
- ◆ [Fonction sans objet pour hyperwave](#)
- ◆ [Fonctionnement des expressions rationnelles](#)
- ◆ [Force l'affichage lorsque ncurses_slk_noutrefresh est utilisé](#)
- ◆ [Force l'exécution d'une requête sur le maître dans une configuration maître/esclave](#)
- ◆ [Force l'exécution d'une requête sur le maître dans une configuration maître/esclave \(master/slave\)](#)
- ◆ [Force l'écriture de toutes les données compressées](#)
- ◆ [Formate un nombre pour l'affichage](#)
- ◆ [Formate une date/heure gmt/cut](#)
- ◆ [Formate une date/heure gmt/cut en fonction des paramètres locaux](#)
- ◆ [Formate une date/heure locale](#)
- ◆ [Formate une date/heure locale avec les options locales](#)
- ◆ [Formate une date/heure locale en tant qu'entier](#)
- ◆ [Formate une ligne en csv et l'écrit dans un fichier](#)
- ◆ [Fusionne trois chaînes en une seule](#)
- ◆ [Fusionne trois fichiers en un seul](#)
- ◆ [Fusionne un ou plusieurs tableaux](#)
- **G**
- - ◆ [Gestion des colonnes de type long](#)
 - ◆ [Gestionnaire de sortie pour maîtriser le jeu de caractères de sortie](#)
 - ◆ [Gets the last sqlstate error code generated](#)
 - ◆ [Gère une requête soap](#)
 - ◆ [Génère le code xml d'une requête de méthode](#)
 - ◆ [Génère le code xml pour une valeur php](#)
 - ◆ [Génère le contexte de débogage](#)
 - ◆ [Génère le formulaire html de demande pour le paiement](#)
 - ◆ [Génère le remerciement lors de la confirmation du paiement](#)
 - ◆ [Génère un buffer openal](#)
 - ◆ [Génère un identifiant unique](#)
 - ◆ [Génère un identifiant unique global](#)
 - ◆ [Génère un message dans l'historique système](#)
 - ◆ [Génère une chaîne de requête en encodage url](#)
 - ◆ [Génère une clé](#)
 - ◆ [Génère une clé de hachage pour les vérifications de certificat de client ssl](#)
 - ◆ [Génère une csr](#)
 - ◆ [Génère une image au format gd, vers le navigateur ou un fichier](#)
 - ◆ [Génère une image au format gd2, vers le navigateur ou un fichier](#)
 - ◆ [Génère une image au format xbm](#)
 - ◆ [Génère une nouvelle clé privée](#)
 - ◆ [Génère une ressource de source](#)
 - ◆ [Génère une valeur aléatoire](#)
 - ◆ [Génère une valeur aléatoire \(meilleure méthode\)](#)
 - ◆ [Générateur de congruence combinée linéaire](#)

- H
- - ◆ Homothétie
- I
- - ◆ Identifiant d'objet de l'objet dans l'ancrage
 - ◆ Identifiant d'objet des parents
 - ◆ Identifiant d'objets des groupes fils
 - ◆ Identifiant des documents fils d'un groupe
 - ◆ Identifiants des ancrages d'un document
 - ◆ Identification avec une connexion ftp
 - ◆ Identification en tant que "none"
 - ◆ Identification en utilisant une clé d'hôte publique
 - ◆ Identification en utilisant une clé publique
 - ◆ Identification via ssh en utilisant un mot de passe en clair
 - ◆ Identifie les attributs d'un élément
 - ◆ Identifie un utilisateur
 - ◆ Ignore les actions si le cadre flash n'est pas chargé
 - ◆ Ignore les mots de moins de n caractères
 - ◆ Importe de nouvelles définitions de méthodes de classes depuis un fichier
 - ◆ Importe les variables dans la table des symboles
 - ◆ Importe les variables de get/post/cookie dans l'environnement global
 - ◆ Importe un fichier dans lob oracle
 - ◆ Importe un noeud dans le document courant
 - ◆ Importe un objet de grande taille depuis un fichier
 - ◆ Importe une feuille de style
 - ◆ Imprime un texte avec des options
 - ◆ Imprime un texte à la position courante
 - ◆ Incline (abscisses)
 - ◆ Incline (ordonnées)
 - ◆ Incline le système de coordonnées
 - ◆ Incline suivant les x
 - ◆ Incline suivant les x relativement
 - ◆ Incline suivant les y
 - ◆ Incline suivant les y relativement
 - ◆ Incline une police de caractères postscript
 - ◆ Incrément la valeur d'un élément
 - ◆ Incrémente le compteur de références (déconseillé)
 - ◆ Incrémente le générateur donné et retourne sa nouvelle valeur
 - ◆ Incrémente une valeur dans une session
 - ◆ Indique comment afficher le document à l'utilisateur
 - ◆ Indique de quoi est capable le navigateur client
 - ◆ Indique la quantité de mémoire utilisée par php
 - ◆ Indique la taille des données à lire dans une colonne de grande taille
 - ◆ Indique le nombre de lignes qui doivent être lues à l'avance par oracle
 - ◆ Indique s'il reste des lignes sqlite à lire
 - ◆ Indique si ce noeud de texte contient des espaces blancs
 - ◆ Indique si deux noeuds sont identiques
 - ◆ Indique si l'entrée courante de simplexml est un objet
 - ◆ Indique si l'internaute a abandonné la connexion http
 - ◆ Indique si la fin d'un fichier compressé est atteinte
 - ◆ Indique si le document est un document xhtml
 - ◆ Indique si le document est un document xml générique (non html/xhtmll)
 - ◆ Indique si le fichier a été téléchargé par http post
 - ◆ Indique si le fichier est exécutable
 - ◆ Indique si le fichier est un dossier

- ◆ Indique si le fichier est un lien symbolique
- ◆ Indique si le fichier est un véritable fichier
- ◆ Indique si le script a expiré
- ◆ Indique si le thread safety est donné ou non
- ◆ Indique si le thread safety est donné ou non
- ◆ Indique si les en-têtes http ont déjà été envoyés
- ◆ Indique si php utilise exslt
- ◆ Indique si un algorithme fonctionne par blocs
- ◆ Indique si un fichier est autorisé en lecture
- ◆ Indique si un fichier est autorisé en écriture
- ◆ Indique si un mode fonctionne par blocs
- ◆ Indique si un mode travaille par blocs
- ◆ Indique si un nombre est fini
- ◆ Indique si une fonction est définie
- ◆ Indique si une valeur appartient à un tableau
- ◆ Indique si une valeur est un nombre
- ◆ Indique si une valeur existe
- ◆ Indique si une variable est un scalaire
- ◆ Indique si une variable vaut null
- ◆ Informations à propos d'une connexion
- ◆ Informe readline que le curseur est passé à une nouvelle ligne
- ◆ Initialise ccvs pour utilisation
- ◆ Initialise l'interface et le terminal de callback de readline, affiche le prompt et retourne immédiatement
- ◆ Initialise la bibliothèque payflow pro
- ◆ Initialise la récupération d'un jeu de résultats
- ◆ Initialise la structure globale d'un flux mcal
- ◆ Initialise le générateur de nombres aléatoires
- ◆ Initialise le serveur mysql intégré
- ◆ Initialise les fonctions de clés soft label
- ◆ Initialise maxdb et retourne une ressource pour l'utiliser avec maxdb_real_connect
- ◆ Initialise mysqli et retourne une ressource à utiliser avec mysqli_real_connect()
- ◆ Initialise ncurses
- ◆ Initialise tous les buffers nécessaires
- ◆ Initialise toutes les constantes liées au syslog
- ◆ Initialise un jeu de résultats
- ◆ Initialise un nouveau pattern pdf
- ◆ Initialise un nouveau pointeur vide de lob/file oracle
- ◆ Initialise un nouveau template pdf
- ◆ Initialise un publickey subsystem (sous-système de clé publique)
- ◆ Initialise un serveur embarqué
- ◆ Initialise un sous-système sftp
- ◆ Initialise un écran depuis un fichier de sauvegarde
- ◆ Initialise une commande mysql
- ◆ Initialise une interface audio openal
- ◆ Initialise une nouvelle collection oracle
- ◆ Initialise une nouvelle page de document pdf
- ◆ Initialise une procédure stockée ms sql server locale ou distante
- ◆ Initialise une requête et retourne une ressource à utiliser avec maxdb_stmt_prepare
- ◆ Initialise une session
- ◆ Initialise une session curl
- ◆ Initialise une valeur aléatoire (meilleure méthode)
- ◆ Inspecte le résultat extended services
- ◆ Installe un gestionnaire de signaux

- ◆ Insère des lignes dans une table à partir d'un tableau
- ◆ Insère des lignes devant la ligne courante en scrollant vers le bas (des nombres négatifs donneront un scroll vers le haut)
- ◆ Insère ou modifie une variable de la mémoire partagée
- ◆ Insère un caractère et décale le reste de la ligne vers la droite, y compris le caractère courant
- ◆ Insère un document
- ◆ Insère un document dans un groupe
- ◆ Insère un groupe
- ◆ Insère un nouveau noeud fils
- ◆ Insère un nouvel attribut hyperwave
- ◆ Insère un nouvel objet
- ◆ Insère un nouvel objet de type ancre
- ◆ Insère un nouvel objet de type collection
- ◆ Insère un nouvel objet de type document
- ◆ Insère un objet record
- ◆ Insère un retour à la ligne html à chaque nouvelle ligne
- ◆ Insère un tableau dans une table
- ◆ Insère une chaîne à la position courante, et décale le reste de la chaîne à droite
- ◆ Insère une chaîne à la position spécifiée d'unité 16-bit
- ◆ Insère une entrée dba
- ◆ Insère une ligne et décale le reste de l'écran vers le bas
- ◆ Insère une valeur
- ◆ Insère uniquement des ancrs dans du texte
- ◆ Interroge l'autre extrémité de la communication
- ◆ Interroge la socket locale
- ◆ Interroge le serveur sur les informations who
- ◆ Intègre des données binaires iptc dans une image jpeg
- ◆ Inverse l'ordre des éléments d'un tableau
- ◆ Inverse modulo
- ◆ Inverse une chaîne
- ◆ Invite un utilisateur à rejoindre un canal
- J
- - ◆ Joue l'animation flash à partir du cadre courant
 - ◆ Joue le cadre flash spécifié
 - ◆ Joue un cadre puis stoppe
- L
- - ◆ L'itérateur secondaire actif courant
 - ◆ La classe dotnet
 - ◆ La plus grand valeur aléatoire possible
 - ◆ La plus grande valeur
 - ◆ La plus petite valeur
 - ◆ Lance une alerte et un callstack
 - ◆ Lance une erreur, un callstack et sort
 - ◆ Le jour de l'année
 - ◆ Le jour de la semaine
 - ◆ Le temps maximal que l'api tentera une connexion à mcve
 - ◆ Lecture binaire d'un fichier bzip2
 - ◆ Lecture de fichier compressé binaire
 - ◆ Lecture du fichier en mode binaire
 - ◆ Lecture du pointeur de fiche courante (cursorname)
 - ◆ Libère la mémoire
 - ◆ Libère la mémoire allouée pour ispell par mnogosearch
 - ◆ Libère la mémoire associée avec la connectivité ip/ssl

- ◆ [Libère la mémoire associée à un résultat](#)
- ◆ [Libère la mémoire associée à un résultat](#)
- ◆ [Libère la mémoire des ressources ms sql server](#)
- ◆ [Libère la mémoire des résultats sotckés pour un gestionnaire de requête donné](#)
- ◆ [Libère la mémoire du résultat](#)
- ◆ [Libère la mémoire occupée par une police postscript type 1](#)
- ◆ [Libère la mémoire réservée par une requête préparée](#)
- ◆ [Libère le résultat de la mémoire](#)
- ◆ [Libère le résultat de la mémoire](#)
- ◆ [Libère le résultat de la mémoire](#)
- ◆ [Libère le résultat mysql de la mémoire](#)
- ◆ [Libère les ressources](#)
- ◆ [Libère les ressources associées avec un objet de collection](#)
- ◆ [Libère les ressources associées à un résultat](#)
- ◆ [Libère les ressources prises par un certificat](#)
- ◆ [Libère les ressources prises par un résultat](#)
- ◆ [Libère les ressources sesam](#)
- ◆ [Libère les ressources utilisées par un résultat ovrimos](#)
- ◆ [Libère tous les verrous posés par le client](#)
- ◆ [Libère toutes les ressources réservées par un résultat oracle](#)
- ◆ [Libère toutes les ressurses](#)
- ◆ [Libère un gestionnaire d'événements ibase](#)
- ◆ [Libère un résultat ibase](#)
- ◆ [Libère un résultat mnogosearch](#)
- ◆ [Libère un résultat sybase de la mémoire](#)
- ◆ [Libère un segment de mémoire partagée](#)
- ◆ [Libère un sémaphore](#)
- ◆ [Libère un verrou en écriture posé sur une relation](#)
- ◆ [Libère une clé privée](#)
- ◆ [Libère une commande ms sql server de la mémoire](#)
- ◆ [Lie des variables à un jeu de résultats](#)
- ◆ [Lie des variables à une requête mysql](#)
- ◆ [Lie les variables à une requête préparée en tant que paramètres](#)
- ◆ [Lie les variables à une requête préparée pour stockage du résultat](#)
- ◆ [Lie un nom à une socket](#)
- ◆ [Lie un paramètre à un nom de variable spécifique](#)
- ◆ [Lie une colonne à une variable php](#)
- ◆ [Lie une variable php à un paramètre oracle](#)
- ◆ [Linéarise une variable](#)
- ◆ [Lire la taille du bloc de mémoire partagée](#)
- ◆ [Lire le nom du groupe](#)
- ◆ [Liste des objets ids des objets fils](#)
- ◆ [Liste des objets records des objets fils](#)
- ◆ [Liste des utilisateurs actuellement identifiés](#)
- ◆ [Liste l'historique](#)
- ◆ [Liste le sujet et le nombre d'utilisateurs d'un ou plusieurs canaux](#)
- ◆ [Liste les bases de données disponibles sur le serveur mysql](#)
- ◆ [Liste les bases de données fbsql](#)
- ◆ [Liste les bases de données msql sur un serveur](#)
- ◆ [Liste les boîtes aux lettres, et retourne les détails de chacune](#)
- ◆ [Liste les champs d'un résultat frontbase](#)
- ◆ [Liste les champs d'une table](#)
- ◆ [Liste les champs d'une table mysql](#)
- ◆ [Liste les champs informix sql](#)

- ◆ [Liste les clés publiques actuellement autorisées](#)
- ◆ [Liste les clés étrangères](#)
- ◆ [Liste les colonnes d'une table](#)
- ◆ [Liste les colonnes et leurs droits associés](#)
- ◆ [Liste les colonnes utilisées dans une clé primaire](#)
- ◆ [Liste les fichiers et dossiers dans un dossier](#)
- ◆ [Liste les filtres enregistrés](#)
- ◆ [Liste les fonctions d'une extension](#)
- ◆ [Liste les gestionnaires d'affichage utilisés](#)
- ◆ [Liste les gestionnaires dba disponibles](#)
- ◆ [Liste les gestionnaires de flux](#)
- ◆ [Liste les gestionnaires de transports de sockets disponibles](#)
- ◆ [Liste les paramètres des procédures](#)
- ◆ [Liste les processus mysql](#)
- ◆ [Liste les procédures stockées](#)
- ◆ [Liste les propriétés des champs sql](#)
- ◆ [Liste les sessions contenant une variable](#)
- ◆ [Liste les tables d'une base de données mysql](#)
- ◆ [Liste les tables d'une source](#)
- ◆ [Liste les tables dans une base de données frontbase](#)
- ◆ [Liste les tables et leurs privilèges](#)
- ◆ [Liste les tables msql sur une base de données](#)
- ◆ [Liste les types de données supportés par une source](#)
- ◆ [Liste les éléments de l'archive rar](#)
- ◆ [Liste tous les algorithmes de chiffrement supportés](#)
- ◆ [Liste tous les fichiers de bases de données dba ouverts](#)
- ◆ [Liste tous les modes de chiffrement supportés](#)
- ◆ [Liste toutes les ancrs de destination](#)
- ◆ [Liste toutes les boîtes aux lettres enregistrées](#)
- ◆ [Liste toutes les boîtes aux lettres souscrites](#)
- ◆ [Liste toutes les catégories soeurs d'une catégorie](#)
- ◆ [Liste toutes les classes définies](#)
- ◆ [Liste toutes les fonctions définies](#)
- ◆ [Liste toutes les sessions](#)
- ◆ [Liste toutes les variables définies](#)
- ◆ [Lit len octets de données dans un blob ibase ouvert](#)
- ◆ [Lit certains paramètres](#)
- ◆ [Lit certains paramètres numériques](#)
- ◆ [Lit des données d'une socket](#)
- ◆ [Lit des données dans un résultat](#)
- ◆ [Lit des données depuis une socket, connectée ou pas](#)
- ◆ [Lit des données à partir d'un bloc](#)
- ◆ [Lit des informations sur le fichier](#)
- ◆ [Lit des informations sur une colonne dans un résultat, et retourne un objet](#)
- ◆ [Lit des informations à propos de la dernière requête mysql](#)
- ◆ [Lit des options passées dans la ligne de commande](#)
- ◆ [Lit des paramètres numériques dans le document pdf d'entrée](#)
- ◆ [Lit des paramètres textuels dans le document pdi](#)
- ◆ [Lit des résultats depuis une requête mysql préparée dans des variables liées](#)
- ◆ [Lit différentes options pour la connexion ftp courante](#)
- ◆ [Lit directement le type de colonne oracle](#)
- ◆ [Lit et analyse un fichier mib dans l'arbre actif mib](#)
- ◆ [Lit et crée des classes depuis un fichier exe bcompiler](#)
- ◆ [Lit et crée les classes depuis un descripteur de fichier](#)

- ◆ Lit et crée les classes depuis un fichier compressé bzipé
- ◆ Lit et modifie les options des fonctions d'expression rationnelle à support de caractères multi-octets
- ◆ Lit et vérifie un fichier
- ◆ Lit et/ou modifie l'identifiant courant de session
- ◆ Lit et/ou modifie la durée de vie de la session
- ◆ Lit et/ou modifie le chemin de sauvegarde des sessions
- ◆ Lit et/ou modifie le limiteur de cache de session
- ◆ Lit et/ou modifie le module de session courant
- ◆ Lit et/ou modifie le nom de la session
- ◆ Lit l'apparence d'un champ
- ◆ Lit l'attribut suivant
- ◆ Lit l'en-tête du message
- ◆ Lit l'encodage du client
- ◆ Lit l'heure locale
- ◆ Lit l'historique
- ◆ Lit l'identifiant de processus du serveur postgresql
- ◆ Lit l'index de la couleur la plus proche d'une couleur spécifiée avec sa teinte, blanc et noir
- ◆ Lit l'os d'un élément d'archive rar
- ◆ Lit l'échelle d'un champ
- ◆ Lit l'échelle d'une colonne oracle
- ◆ Lit la clé dba suivante
- ◆ Lit la clé suivante
- ◆ Lit la configuration de sécurité du dossier
- ◆ Lit la configuration du cookie de session
- ◆ Lit la configuration interne de l'extension mbstring
- ◆ Lit la configuration locale
- ◆ Lit la configuration tidy courante
- ◆ Lit la date de dernière modification d'un élément d'archive rar
- ◆ Lit la dernière erreur générée par une socket
- ◆ Lit la dernière erreur libxml
- ◆ Lit la dernière exception java
- ◆ Lit la dernière ligne d'une relation
- ◆ Lit la ligne courante dans une relation
- ◆ Lit la ligne précédente dans la relation
- ◆ Lit la ligne suivante dans la relation
- ◆ Lit la liste des boîtes aux lettres
- ◆ Lit la liste des boîtes aux lettres, et y recherche une chaîne
- ◆ Lit la longueur d'un champ
- ◆ Lit la longueur d'un champ ms sql server
- ◆ Lit la méthode de compression d'un élément d'archive rar
- ◆ Lit la méthode de compression utilisée sur un dossier d'archives
- ◆ Lit la position courante du pointeur de lecture
- ◆ Lit la première clé
- ◆ Lit la première clé dba
- ◆ Lit la première ligne d'un résultat sqlite sous forme de chaîne
- ◆ Lit la première ligne d'une relation
- ◆ Lit la prochaine entrée
- ◆ Lit la prochaine entrée dans une archive zip
- ◆ Lit la prochaine ligne dans le résultat oracle
- ◆ Lit la prochaine ligne de résultat sqlite dans un tableau
- ◆ Lit la précision d'une colonne oracle
- ◆ Lit la référence suivante

- ◆ [Lit la réponse de l'api muscat](#)
- ◆ [Lit la somme de contrôle d'un élément d'archive rar](#)
- ◆ [Lit la structure d'un message](#)
- ◆ [Lit la structure d'une section du corps d'un mail](#)
- ◆ [Lit la taille compressée d'un dossier d'archives](#)
- ◆ [Lit la taille compressée d'un élément d'archive rar](#)
- ◆ [Lit la taille d'une colonne oracle](#)
- ◆ [Lit la taille de chaque colonne d'un résultat](#)
- ◆ [Lit la taille du champ mysql](#)
- ◆ [Lit la taille décompressée d'un dossier d'archives](#)
- ◆ [Lit la taille décompressée d'un élément d'archive rar](#)
- ◆ [Lit la taille logique de la police](#)
- ◆ [Lit la totalité d'un fichier compressé](#)
- ◆ [Lit la valeur courante de l'option quick_print de la bibliothèque ucd](#)
- ◆ [Lit la valeur d'un champ mysql](#)
- ◆ [Lit la valeur d'une colonne dans un résultat sqlite](#)
- ◆ [Lit la valeur d'une option de configuration](#)
- ◆ [Lit la valeur d'une propriété d'un composant com](#)
- ◆ [Lit la valeur de la clé /encoding](#)
- ◆ [Lit la valeur de la clé /f](#)
- ◆ [Lit la valeur de la clé /status](#)
- ◆ [Lit la valeur de la directive de configuration include_path](#)
- ◆ [Lit la valeur des options pour un flux/gestionnaire/contexte](#)
- ◆ [Lit la valeur rgb d'une couleur](#)
- ◆ [Lit la version courante du moteur zend](#)
- ◆ [Lit la version de l'api mnogosearch](#)
- ◆ [Lit la version de lotus notes](#)
- ◆ [Lit la version utilisée pour créer le dernier élément rar](#)
- ◆ [Lit le bytecode d'une classe et revient à une fonction utilisateur](#)
- ◆ [Lit le caractère et ses attributs à la position courante](#)
- ◆ [Lit le caractère qui se fait effacer](#)
- ◆ [Lit le chemin de la catégorie courante](#)
- ◆ [Lit le code d'erreur généré par la dernière opération](#)
- ◆ [Lit le contenu](#)
- ◆ [Lit le contenu courant du buffer de sortie puis l'efface](#)
- ◆ [Lit le contenu d'un fichier dans un dossier](#)
- ◆ [Lit le contenu d'une colonne ovrimos](#)
- ◆ [Lit le contexte par défaut des flux](#)
- ◆ [Lit le corps d'un message](#)
- ◆ [Lit le dernier code d'erreur](#)
- ◆ [Lit le dernier identifiant généré par une requête insert](#)
- ◆ [Lit le dernier message d'erreur](#)
- ◆ [Lit le dernier message d'erreur sur la connexion](#)
- ◆ [Lit le dernier message d'erreur vpopmail](#)
- ◆ [Lit le dernier numéro d'erreur généré par la dernière fonction posix qui a échoué](#)
- ◆ [Lit le dernier segment de chaîne multi-octets qui correspond au masque](#)
- ◆ [Lit le dn d'une entrée](#)
- ◆ [Lit le fichier et renvoie le résultat dans un tableau](#)
- ◆ [Lit le jeu de caractères courant](#)
- ◆ [Lit le mappage de script dans un dossier virtuel donné](#)
- ◆ [Lit le message d'erreur associé à un numéro d'erreur posix](#)
- ◆ [Lit le message d'erreur associé à un résultat](#)
- ◆ [Lit le message d'erreur de l'analyseur xml](#)
- ◆ [Lit le message sql notify](#)

- ◆ [Lit le mime-type d'un type d'image](#)
- ◆ [Lit le nom d'en-tête exif d'un index](#)
- ◆ [Lit le nom d'un champ](#)
- ◆ [Lit le nom d'un champ ms sql server](#)
- ◆ [Lit le nom d'un champ ms sql server](#)
- ◆ [Lit le nom d'un dossier d'archives](#)
- ◆ [Lit le nom d'un élément de code source](#)
- ◆ [Lit le nom de l'élément d'archive rar](#)
- ◆ [Lit le nom de la colonne](#)
- ◆ [Lit le nom de la table d'origine d'un champ](#)
- ◆ [Lit le nom de la table d'un champ](#)
- ◆ [Lit le nom de la table du champ mysql](#)
- ◆ [Lit le nom de la table qui contient un champ](#)
- ◆ [Lit le nom des sockets locale ou distante](#)
- ◆ [Lit le nom du champ fdf suivant](#)
- ◆ [Lit le nom du champ mysql](#)
- ◆ [Lit le nom du chiffrement utilisé](#)
- ◆ [Lit le nombre de champs dans un résultat](#)
- ◆ [Lit le nombre de lignes affectées par la dernière requête](#)
- ◆ [Lit le nombre de lignes affectées par une requête sesam immédiate](#)
- ◆ [Lit le nombre de lignes dans un résultat](#)
- ◆ [Lit le nombre total de documents dans les bases mnogosearch](#)
- ◆ [Lit le numéro de version de l'api fdf](#)
- ◆ [Lit le numéro de version de la bibliothèque xml](#)
- ◆ [Lit le numéro de version de la bibliothèque xslt](#)
- ◆ [Lit le reste d'un flux dans une chaîne](#)
- ◆ [Lit le statut courant d'une base de données](#)
- ◆ [Lit le statut de la connexion postgresql](#)
- ◆ [Lit le statut du buffer de sortie](#)
- ◆ [Lit le statut du résultat](#)
- ◆ [Lit le tableau d'erreurs](#)
- ◆ [Lit le tampon contenant le fichier pdf généré](#)
- ◆ [Lit le type d'une colonne](#)
- ◆ [Lit le type de champ mysql](#)
- ◆ [Lit les attributs d'un champ fdf](#)
- ◆ [Lit les attributs d'un objet hyperwave](#)
- ◆ [Lit les attributs d'un élément d'archive rar](#)
- ◆ [Lit les attributs d'une entrée](#)
- ◆ [Lit les données d'un résultat](#)
- ◆ [Lit les données dans un fichier, applique l'encodage et envoie le résultat à destfp](#)
- ◆ [Lit les données de la session](#)
- ◆ [Lit les données dns associées à un hôte](#)
- ◆ [Lit les données liées à une clé dba](#)
- ◆ [Lit les droits du serveur](#)
- ◆ [Lit les en-têtes exif dans les images jpeg ou tiff](#)
- ◆ [Lit les en-têtes et données méta des flux](#)
- ◆ [Lit les informations concernant un processus ouvert par proc_open](#)
- ◆ [Lit les informations d'attribut](#)
- ◆ [Lit les informations d'un champ sybase](#)
- ◆ [Lit les informations de configuration de l'imprimante](#)
- ◆ [Lit les informations du client mysql](#)
- ◆ [Lit les informations détaillant un transfert curl](#)
- ◆ [Lit les informations sur l'hôte mysql](#)
- ◆ [Lit les informations sur le champ](#)

- ◆ [Lit les informations sur le client mysql](#)
- ◆ [Lit les informations sur le protocole mysql](#)
- ◆ [Lit les informations sur le serveur mysql](#)
- ◆ [Lit les informations sur les transferts actuels](#)
- ◆ [Lit les informations sur un champ ibase](#)
- ◆ [Lit les informations sur un fichier à partir d'un pointeur de fichier](#)
- ◆ [Lit les informations sur une image](#)
- ◆ [Lit les informations à propos de la boîte aux lettres courante](#)
- ◆ [Lit les métadonnées de la table postgresql](#)
- ◆ [Lit les noms des bases de données](#)
- ◆ [Lit les octets dans un fichier](#)
- ◆ [Lit les options associées à une colonne de résultat](#)
- ◆ [Lit les options d'un analyseur xml](#)
- ◆ [Lit les options de champ msql](#)
- ◆ [Lit les options de la socket](#)
- ◆ [Lit les paramètres de résultats mnogosearch](#)
- ◆ [Lit les quotas de chaque utilisateur](#)
- ◆ [Lit les quotas des boîtes aux lettres](#)
- ◆ [Lit les statistiques du serveur](#)
- ◆ [Lit les éléments par nom de balise](#)
- ◆ [Lit les événements souris](#)
- ◆ [Lit n bytes d'un objet slob](#)
- ◆ [Lit ou définit le mot de passe à utiliser avec une connexion](#)
- ◆ [Lit ou définit le nom de l'hôte à utiliser avec une connexion](#)
- ◆ [Lit ou définit le nom de la base de données utilisée avec une connexion](#)
- ◆ [Lit ou définit le nom de login à utiliser avec une connexion](#)
- ◆ [Lit tous les en-têtes de requête http sur un serveur netscape](#)
- ◆ [Lit tous les en-têtes de réponse http sur netscape serveur](#)
- ◆ [Lit tout le fichier compressé](#)
- ◆ [Lit tout un fichier dans une chaîne](#)
- ◆ [Lit toute une ligne de résultat dans un tableau](#)
- ◆ [Lit toute une ligne de résultat dans un tableau associatif](#)
- ◆ [Lit toutes les entrées du résultat](#)
- ◆ [Lit toutes les lignes d'alias d'un domaine](#)
- ◆ [Lit toutes les lignes d'un alias de domaine](#)
- ◆ [Lit toutes les lignes d'un résultat](#)
- ◆ [Lit toutes les lignes d'un résultat oracle](#)
- ◆ [Lit toutes les lignes d'un tableau, et le met sous la forme d'un tableau html](#)
- ◆ [Lit toutes les valeurs binaires d'une entrée](#)
- ◆ [Lit toutes les valeurs d'une entrée ldap](#)
- ◆ [Lit toutes les valeurs de configuration](#)
- ◆ [Lit un blob dans une base de données](#)
- ◆ [Lit un caractère dans un fichier](#)
- ◆ [Lit un caractère dans un fichier compressé](#)
- ◆ [Lit un caractère depuis le clavier \(fenêtre\)](#)
- ◆ [Lit un caractère et informe l'interface de callback readline lorsqu'une ligne est reçue](#)
- ◆ [Lit un caractère sur le clavier](#)
- ◆ [Lit un champ de résultat mnogosearch](#)
- ◆ [Lit un champ de résultat uodbc](#)
- ◆ [Lit un clob dans une base de données](#)
- ◆ [Lit un document fdf à partir d'une chaîne de caractères](#)
- ◆ [Lit un enregistrement dans une base dbase](#)
- ◆ [Lit un enregistrement dans une base dbase, sous la forme d'un tableau associatif](#)
- ◆ [Lit un fichier sur un serveur ftp, et l'écrit dans un fichier \(non bloquant\)](#)

- ◆ Lit un fichier sur un serveur ftp, et l'écrit dans un fichier (non bloquant)
- ◆ Lit un objet de grande taille
- ◆ Lit un objet de grande taille en totalité
- ◆ Lit un résultat postgresql asynchrone
- ◆ Lit un sommaire des en-têtes de messages
- ◆ Lit un tableau de variables msession
- ◆ Lit un élément dans l'archive rar
- ◆ Lit une chaîne aléatoire
- ◆ Lit une chaîne depuis le terminal
- ◆ Lit une clé privée
- ◆ Lit une entrée
- ◆ Lit une entrée du dossier
- ◆ Lit une image depuis un fichier
- ◆ Lit une image pour placement futur
- ◆ Lit une ligne
- ◆ Lit une ligne d'un résultat oracle sous forme d'objet
- ◆ Lit une ligne d'un résultat oracle sous forme de tableau
- ◆ Lit une ligne d'un résultat oracle sous forme de tableau
- ◆ Lit une ligne d'un résultat oracle sous forme de tableau associatif
- ◆ Lit une ligne dans un fichier compressé
- ◆ Lit une ligne dans un fichier compressé, et supprime les balises html
- ◆ Lit une ligne dans un flux
- ◆ Lit une ligne dans un résultat dbx ayant l'option dbx_result_unbuffered activée
- ◆ Lit une ligne dans un résultat ovrimos
- ◆ Lit une ligne dans un résultat ovrimos
- ◆ Lit une ligne dans un tableau
- ◆ Lit une ligne dans un tableau
- ◆ Lit une ligne dans un tableau associatif
- ◆ Lit une ligne dans une base interbase
- ◆ Lit une ligne dans une base interbase dans un objet
- ◆ Lit une ligne de résultat
- ◆ Lit une ligne de résultat dans un objet
- ◆ Lit une ligne de résultat dans un tableau associatif
- ◆ Lit une ligne de résultat ms sql dans un tableau
- ◆ Lit une ligne de résultat ms sql dans un tableau numérique
- ◆ Lit une ligne de résultat mysql dans un tableau associatif
- ◆ Lit une ligne de résultat oracle dans un tableau
- ◆ Lit une ligne de résultat postgresql dans un objet
- ◆ Lit une ligne de résultat postgresql dans un tableau
- ◆ Lit une ligne de résultat postgresql sous forme de tableau numérique
- ◆ Lit une ligne de résultat sous forme d'objet
- ◆ Lit une ligne de résultat sous forme de tableau numérique
- ◆ Lit une ligne de résultat sqlite dans un tableau
- ◆ Lit une ligne de résultat sybase sous forme de tableau associatif
- ◆ Lit une ligne de résultat, et la place dans un tableau
- ◆ Lit une ligne sous la forme d'un objet
- ◆ Lit une ligne sous la forme d'un tableau
- ◆ Lit une partie d'un lob oracle
- ◆ Lit une représentation exportable de la clé dans une chaîne ou un fichier
- ◆ Lit une sous-chaîne
- ◆ Lit une valeur
- ◆ Lit une valeur dans un résultat
- ◆ Lit une valeur dans un tableau de valeur d'un champ fdf
- ◆ Lit une valeur dans une session

- ◆ Lit une variable dans la mémoire partagée
- ◆ Lit/modifie diverses variables internes
- ◆ Lit/modifie l'encodage d'affichage
- ◆ Lit/modifie l'encodage interne
- ◆ Lit/modifie l'ordre de détection des encodages
- ◆ Lit/modifie le chemin courant de base
- ◆ Lit/modifie le langage courant
- ◆ Lit/modifie les caractères de substitution
- ◆ Lit/écrit la valeur courante d'une option
- ◆ Logarithme en base 10
- ◆ Logarithme naturel (népérien)
- ◆ Lève tous les verrous sur les lignes d'une relation
- ◆ Lève un verrou en écriture sur une ligne
- ◆ Lève un verrou exclusif posé sur une relation
- **M**
- - ◆ Marque la note note_id comme lue par l'utilisateur user_name
 - ◆ Marque la note note_id comme non lue pour l'utilisateur user_name
 - ◆ Marque le fichier pour l'effacement, dans la boîte aux lettres courante
 - ◆ Marque une pause dans la source
 - ◆ Message d'erreur mnogosearch
 - ◆ Met en majuscule la première lettre de tous les mots
 - ◆ Met la socket en mode bloquant
 - ◆ Met le premier caractère en majuscule
 - ◆ Met le terminal en mode semi-retardé
 - ◆ Met tous les caractères en majuscules
 - ◆ Met tous les caractères en minuscules
 - ◆ Met un nombre au format monétaire
 - ◆ Met à jour les informations stockées dans un tag id3
 - ◆ Modifie certains paramètres
 - ◆ Modifie certains paramètres numériques
 - ◆ Modifie des informations dans la file de messages
 - ◆ Modifie l'action d'un formulaire
 - ◆ Modifie l'action javascript d'un champ
 - ◆ Modifie l'alignement du texte
 - ◆ Modifie l'encodage des caractères
 - ◆ Modifie l'espace de noms d'un noeud
 - ◆ Modifie l'espacement de lignes
 - ◆ Modifie l'espacement de police
 - ◆ Modifie l'identifiant d'utilisateur réel (uid) du processus courant
 - ◆ Modifie l'image utilisée pour le carrelage
 - ◆ Modifie l'indentation de la première ligne
 - ◆ Modifie l'index d'un champ sybase
 - ◆ Modifie l'origine du système de coordonnées
 - ◆ Modifie l'utilisateur de la connexion à la base de données spécifiée
 - ◆ Modifie l'échelle
 - ◆ Modifie l'échelle de la forme
 - ◆ Modifie l'épaisseur d'un trait
 - ◆ Modifie l'état du curseur
 - ◆ Modifie la brosse pour le dessin des lignes
 - ◆ Modifie la casse d'une chaîne
 - ◆ Modifie la configuration de sécurité du dossier
 - ◆ Modifie la couleur de fond
 - ◆ Modifie la couleur de fond et de devant
 - ◆ Modifie la couleur de la police

- ◆ Modifie la couleur de rastérisation de droite
- ◆ Modifie la couleur de rastérisation de gauche
- ◆ Modifie la couleur du champ texte
- ◆ Modifie la date de modification et de dernier accès d'un fichier
- ◆ Modifie la gestion des colonnes de données binaires
- ◆ Modifie la hauteur de la police courante
- ◆ Modifie la hauteur de la police du champ texte
- ◆ Modifie la ligne courant dans un résultat
- ◆ Modifie la marge de droite
- ◆ Modifie la marge de gauche
- ◆ Modifie la place en profondeur (z-order)
- ◆ Modifie la police du champ
- ◆ Modifie la position dans un objet de grande taille
- ◆ Modifie la position du pointeur de fichier
- ◆ Modifie la position du texte
- ◆ Modifie la position et dessine une ligne verticale avec le caractère donné, de taille maximale donnée
- ◆ Modifie la valeur d'un champ fdf
- ◆ Modifie la valeur d'une option de configuration
- ◆ Modifie la valeur d'une option ldap
- ◆ Modifie la valeur de l'option de configuration tidy
- ◆ Modifie la valeur de la directive de configuration include_path
- ◆ Modifie la vitesse de l'animation
- ◆ Modifie le acl de la boîte aux lettres
- ◆ Modifie le chemin de clipping pdf
- ◆ Modifie le contenu d'un noeud
- ◆ Modifie le contenu d'un objet blob
- ◆ Modifie le contenu d'un objet char
- ◆ Modifie le gid réel du processus courant
- ◆ Modifie le jeu courant de caractères d'encodage
- ◆ Modifie le jeu de caractères pour les entrées/sorties de l'analyseur tidy
- ◆ Modifie le mappage de script pour un dossier virtuel
- ◆ Modifie le mode de blending d'une image
- ◆ Modifie le mode de lecture des lob
- ◆ Modifie le mode par défaut de lecture des valeurs
- ◆ Modifie le mot de passe d'un utilisateur oracle
- ◆ Modifie le mot de passe du principal
- ◆ Modifie le niveau de gris comme couleur de remplissage
- ◆ Modifie le nom d'un noeud
- ◆ Modifie le nom d'une entrée
- ◆ Modifie le nombre total d'images dans l'animation
- ◆ Modifie le numéro de version du fichier fdf
- ◆ Modifie le quota d'un utilisateur virtuel
- ◆ Modifie le quota d'une boîte aux lettres
- ◆ Modifie le ratio de l'objet
- ◆ Modifie le style de ligne de la forme
- ◆ Modifie le sujet (topic) d'un canal
- ◆ Modifie le type d'appel de la fonction win32
- ◆ Modifie le type xmlrpc, base64 ou datetime, pour une chaîne de caractères php
- ◆ Modifie les attributs d'object record
- ◆ Modifie les attributs d'une fenêtre
- ◆ Modifie les attributs donnés
- ◆ Modifie les attributs du terminal et le taux de baud du port série
- ◆ Modifie les dimensions de l'animation

- ◆ Modifie les droits d'un fichier via ftp
- ◆ Modifie les droits du serveur
- ◆ Modifie les droits sur la relation
- ◆ Modifie les flags du canal
- ◆ Modifie les gestionnaires sax de l'analyseur xslt
- ◆ Modifie les informations de localisation
- ◆ Modifie les lignes d'une table
- ◆ Modifie les marges du champ texte flash
- ◆ Modifie les options de la connexion ftp
- ◆ Modifie les options de socket
- ◆ Modifie les paires de couleurs de la fenêtre
- ◆ Modifie les paramètres de l'agent mnogosearch
- ◆ Modifie les paramètres de transaction sesam
- ◆ Modifie les paramètres du cookie de session
- ◆ Modifie les paramètres odbc
- ◆ Modifie les étiquettes de clé de fonction (function key labels)
- ◆ Modifie ou définit le degré de débogage courant
- ◆ Modifie un bit
- ◆ Modifie un niveau de gris comme couleur de dessin et de remplissage
- ◆ Modifie un objet flash
- ◆ Modifie un principal kerberos avec les paramètres donnés
- ◆ Modifie un utilisateur dans une base de données de sécurité (uniquement pour interbase6 ou plus récent)
- ◆ Modifie un événement dans un calendrier mcal
- ◆ Modifie une entrée ldap
- ◆ Modifie une ligne dans une relation
- ◆ Modifie une option d'un champ
- ◆ Modifie une option d'un champ
- ◆ Modifie une option de transfert curl
- ◆ Modifie une ou plusieurs options de connexion yaz
- ◆ Modifie une propriété d'un composant com
- ◆ Modifie une valeur dans la session
- ◆ Modifie une variable de subprocess env apache
- ◆ Modifie/lit le mot de passe dans une base fbsql
- ◆ Modifie/remplace le contenu d'un document
- ◆ Modulo gmp
- ◆ Monitore les changements d'une collection de fichiers dans un dossier
- ◆ Monitore un dossier pour les changements
- ◆ Monitore un fichier régulier pour les changements
- ◆ Multipliation de 2 nombres gmp
- ◆ Multiplie deux nombres de grande taille
- ◆ Multiplie la couleur de transformation
- ◆ Multiplie les valeurs de deux variants et retourne le résultat
- ◆ Mélange les caractères d'une chaîne de caractères
- ◆ Mélange les éléments d'un tableau
- N
- - ◆ Ncurses est en mode endwin, des affichages normaux sont possibles
 - ◆ Ne convertit par les nouvelles lignes et retours chariot en sauts de ligne
 - ◆ Ne vide pas les buffers sur réception des caractères de signaux
 - ◆ Nom de l'utilisateur actuellement identifié
 - ◆ Nombre d'autorisations complètes en attente, retourne un tableau de leur identifiant
 - ◆ Nombre de colonnes dans un résultat
 - ◆ Nombre de colonnes retournées dans une réponse délimitée par virgule
 - ◆ Nombre de lignes dans un résultat

- ◆ Nombre de lignes retournés dans une réponse délimitée par virgule
- ◆ Nombre de transactions dans la file de client
- ◆ Nombre gmp aléatoire
- ◆ Nombre gmp probablement premier
- ◆ Nomme le cadre courant
- ◆ Nomme le champ texte
- ◆ Nomme un objet
- ◆ Normalise le document
- ◆ Normalise le noeud
- ◆ Numéro d'erreur mnogosearch
- ◆ Numéro de colonne
- 0
- - ◆ Object record de hw_document
 - ◆ Objet id de la racine
 - ◆ Objet record d'un groupe d'enfants
 - ◆ Obsolète : ajoute un signet dans la page courante
 - ◆ Obsolète : ajoute une annotation
 - ◆ Obsolète : chiffre/déchiffre des données en mode ecb
 - ◆ Obsolète : configure des pointillés complexes
 - ◆ Obsolète : configure l'espace entre deux mots
 - ◆ Obsolète : configure l'espacement de caractères
 - ◆ Obsolète : configure l'échelle horizontale du texte
 - ◆ Obsolète : configure l'élévation de texte
 - ◆ Obsolète : configure la distance entre deux lignes de texte
 - ◆ Obsolète : configure la durée entre deux pages
 - ◆ Obsolète : configure la matrice de texte
 - ◆ Obsolète : détermine le rendu du texte
 - ◆ Obsolète : gère les polices
 - ◆ Obsolète : lit le nom de la police
 - ◆ Obsolète : ouvre un nouvel objet pdf
 - ◆ Obsolète : ouvre une image gif
 - ◆ Obsolète : ouvre une image jpeg
 - ◆ Obsolète : ouvre une image png
 - ◆ Obsolète : ouvre une image tiff
 - ◆ Obsolète : remplit le champ d'auteur du document
 - ◆ Obsolète : remplit le champ de créateur du document
 - ◆ Obsolète : remplit le champ de mots-clés du document
 - ◆ Obsolète : remplit le champ de sujet du document
 - ◆ Obsolète : remplit le champ de titre du document
 - ◆ Obsolète : retourne la hauteur d'une image
 - ◆ Obsolète : retourne la largeur d'une image
 - ◆ Obsolète : sélectionne la taille et le nom d'une police
 - ◆ Obsolète : termine le chemin courant
 - ◆ Obsolète: charge une police
 - ◆ Obtenir un message d'erreur
 - ◆ Obtient le statut courant du système
 - ◆ Obtient une représentation textuelle de la cause de l'échec de connexion
 - ◆ Obtient une variable subprocess_env d'apache
 - ◆ Opposé de nombre gmp
 - ◆ Optimise une base dba
 - ◆ Options disponibles pour les expressions rationnelles
 - ◆ Ou exclusif logique
 - ◆ Ou logique
 - ◆ Ouvre une nouvelle connexion sur un serveur maxdb

- ◆ [Ouverture d'un fichier ou d'une url](#)
- ◆ [Ouvre des données scellées](#)
- ◆ [Ouvre la connexion au serveur msession](#)
- ◆ [Ouvre la connexion à l'historique système](#)
- ◆ [Ouvre le message msg_number dans la boîte aux lettres et le serveur précisés](#)
- ◆ [Ouvre le message msg_number dans la boîte aux lettres et le serveur spécifiés](#)
- ◆ [Ouvre le module de l'algorithme et du mode à utiliser](#)
- ◆ [Ouvre un blob ibase pour récupérer des parties de données](#)
- ◆ [Ouvre un connexion persistante à un serveur de cache](#)
- ◆ [Ouvre un document fdf](#)
- ◆ [Ouvre un dossier d'archives en lecture](#)
- ◆ [Ouvre un dossier, et récupère un pointeur dessus](#)
- ◆ [Ouvre un fichier compressé avec bzip2](#)
- ◆ [Ouvre un fichier compressé avec gzip](#)
- ◆ [Ouvre un fichier de relation](#)
- ◆ [Ouvre un fichier de relation local](#)
- ◆ [Ouvre un fichier pdf](#)
- ◆ [Ouvre un fichier pour écriture](#)
- ◆ [Ouvre un flux imap vers une boîte aux lettres](#)
- ◆ [Ouvre un nouveau dictionnaire cracklib](#)
- ◆ [Ouvre un nouveau document pdf](#)
- ◆ [Ouvre un nouveau fichier avec les permissions spécifiés](#)
- ◆ [Ouvre un nouveau fichier pdf](#)
- ◆ [Ouvre un nouveau fichier shockwave flash](#)
- ◆ [Ouvre un objet de grande taille postgresql](#)
- ◆ [Ouvre un objet slob](#)
- ◆ [Ouvre un pointeur oracle](#)
- ◆ [Ouvre un tunnel à travers un serveur distant](#)
- ◆ [Ouvre une archive rar](#)
- ◆ [Ouvre une archive zip](#)
- ◆ [Ouvre une base dbase](#)
- ◆ [Ouvre une base de données dba](#)
- ◆ [Ouvre une base de données dbm](#)
- ◆ [Ouvre une base de données paradox](#)
- ◆ [Ouvre une base sqlite et crée un objet pour elle](#)
- ◆ [Ouvre une base sqlite et la crée si elle n'existe pas](#)
- ◆ [Ouvre une connexion avec le serveur memcache](#)
- ◆ [Ouvre une connexion avec un serveur mysql intégré](#)
- ◆ [Ouvre une connexion ftp](#)
- ◆ [Ouvre une connexion ftp sécurisée avec ssl](#)
- ◆ [Ouvre une connexion hyperwave](#)
- ◆ [Ouvre une connexion icap](#)
- ◆ [Ouvre une connexion mcal](#)
- ◆ [Ouvre une connexion mnogosearch avec un document stocké](#)
- ◆ [Ouvre une connexion msq1](#)
- ◆ [Ouvre une connexion oracle](#)
- ◆ [Ouvre une connexion persistante mcal](#)
- ◆ [Ouvre une connexion persistante à oracle](#)
- ◆ [Ouvre une connexion persistante à un serveur frontbase](#)
- ◆ [Ouvre une connexion persistante à un serveur informix](#)
- ◆ [Ouvre une connexion persistante à un serveur ingres](#)
- ◆ [Ouvre une connexion persistante à un serveur ms sql](#)
- ◆ [Ouvre une connexion persistante à un serveur msq1](#)
- ◆ [Ouvre une connexion persistante à un serveur mysql](#)

- ◆ Ouvre une connexion persistante à un serveur oracle
 - ◆ Ouvre une connexion persistante à un serveur sybase
 - ◆ Ouvre une connexion persistante à une base de données dba
 - ◆ Ouvre une connexion persistante à une base de données interbase
 - ◆ Ouvre une connexion persistante à une source de données
 - ◆ Ouvre une connexion sesam
 - ◆ Ouvre une connexion socket internet ou unix
 - ◆ Ouvre une connexion sqlite persistante et crée la base si elle n'existe pas
 - ◆ Ouvre une connexion sur un serveur maxdb
 - ◆ Ouvre une connexion sur un serveur maxdb embarqué
 - ◆ Ouvre une connexion vers le démon fam
 - ◆ Ouvre une connexion à la bibliothèque kadm5 et initialise toutes les informations nécessaires
 - ◆ Ouvre une connexion à un serveur frontbase
 - ◆ Ouvre une connexion à un serveur informix
 - ◆ Ouvre une connexion à un serveur ingres
 - ◆ Ouvre une connexion à un serveur ms sql server
 - ◆ Ouvre une connexion à un serveur mysql
 - ◆ Ouvre une connexion à un serveur mysql
 - ◆ Ouvre une connexion à un serveur mysql
 - ◆ Ouvre une connexion à un serveur sybase
 - ◆ Ouvre une connexion à une base de données
 - ◆ Ouvre une connexion à une base de données interbase
 - ◆ Ouvre une connexion à une imprimante
 - ◆ Ouvre une iamge à partir d'un fichier
 - ◆ Ouvre une image
 - ◆ Ouvre une image contenant des données brutes ccitt
 - ◆ Ouvre une image créée en mémoire par php
 - ◆ Ouvre une image jpeg
 - ◆ Ouvre une socket de connexion internet ou unix
 - ◆ Ouvre une socket de connexion internet ou unix persistante
 - ◆ Ouvre une socket sur un port pour accepter les connexions
- P
- ◆ Paramètre un hyper lien
 - ◆ Parcours un flux étendu de données
 - ◆ Pas de documentation
 - ◆ Passe le terminal en mode brut (raw)
 - ◆ Passe le terminal en mode cooked
 - ◆ Passe une fenêtre en mode standout
 - ◆ Passe à l'image suivante
 - ◆ Patche un fichier avec un diff binaire
 - ◆ Patche un fichier avec un diff unifié
 - ◆ Patche une chaîne avec un diff binaire
 - ◆ Patche une chaîne avec un diff unifié
 - ◆ Performe une disjonction logique sur deux variants
 - ◆ Performe une exclusion logique sur deux variants
 - ◆ Performe une implication sur les bits de deux variants
 - ◆ Performe une négation sur les bits sur un variant
 - ◆ Performe une équivalence de bits de deux variants
 - ◆ Pgcd
 - ◆ Pgcd étendu
 - ◆ Ping la connexion au serveur et reconnecte si elle n'existe plus
 - ◆ Ping la connexion à la base
 - ◆ Ping ou non sur la connexion pour la vérifier

- ◆ Pingue le serveur mysql. et s'y reconnecte au besoin
- ◆ Place l'itérateur à une position donnée
- ◆ Place le point courant pdf
- ◆ Place le pointeur de résultat sqlite au début
- ◆ Place un conteneur invisible en haut de la pile. pour le rendre visible
- ◆ Place un conteneur visible en bas de la pile
- ◆ Place un conteneur visible sur le haut de la pile
- ◆ Place un objet sur la scène
- ◆ Place un point de couleur pdf
- ◆ Place une image dans la page
- ◆ Place une image dans une page
- ◆ Place une image sur la page
- ◆ Place une page dans le document
- ◆ Planifie une alarme pour délivrer un signal
- ◆ Plus grande valeur aléatoire possible
- ◆ Pose un verrou en écriture sur une relation
- ◆ Pose un verrou exclusif sur une relation
- ◆ Pose un verrou sur une ligne
- ◆ Positionne le pointeur de tableau en fin de tableau
- ◆ Positionne un flag sur un message
- ◆ Prend une ou plusieurs valeurs. au hasard dans un tableau
- ◆ Protège les caractères spéciaux d'une chaîne pour l'utiliser dans une requête sql. en prenant en compte le jeu de caractères courant de la connexion
- ◆ Protège les caractères spéciaux d'une commande sql
- ◆ Protège les caractères spéciaux du shell
- ◆ Protège une chaîne de caractères pour l'insérer dans un champ texte
- ◆ Protège une chaîne de caractères pour l'utiliser dans une requête
- ◆ Protège une chaîne de caractères pour utilisation avec sqlite
- ◆ Protège une chaîne de caractères pour utilisation en ligne de commande
- ◆ Protège une chaîne pour insertion dans un champ bytea
- ◆ Protège une chaîne pour l'utiliser dans une requête sql pdo
- ◆ Protège une chaîne pour la passer à mysql_query
- ◆ Provoque une erreur soapserver indiquant une erreur
- ◆ Préapre le prochain résultat depuis multi_query
- ◆ Précise les dossiers à fouiller pour rechercher des polices externes
- ◆ Prépare des données irc pour l'affichage html
- ◆ Prépare et exécute une requête sql
- ◆ Prépare l'annulation
- ◆ Prépare la connexion courante pour l'affichage
- ◆ Prépare le module pour le téléchargement
- ◆ Prépare le prochain résultat d'une requête multiple
- ◆ Prépare un scan
- ◆ Prépare une commande pour l'exécution
- ◆ Prépare une connexion à un serveur z39.50
- ◆ Prépare une expression rationnelle pour effectuer une recherche insensible à la casse
- ◆ Prépare une page
- ◆ Prépare une police pour utilisation ultérieure avec pdf_setfont
- ◆ Prépare une recherche
- ◆ Prépare une requête ibase pour lier les paramètres et l'exécuter ultérieurement
- ◆ Prépare une requête sql avec oracle
- ◆ Prépare une requête sql ovrinos
- ◆ Prépare une requête sql pour exécution
- ◆ Prépare une requête sql pour exécution

- ◆ [Prépare une requête sql pour l'exécution](#)
- ◆ [Prépare une requête sql pour l'exécution](#)
- ◆ [Prépare une requête sql pour l'exécution](#)
- ◆ [Prépare une requête z39.50 item order avec le package ill-request](#)
- ◆ [Prépare une requête à l'exécution et retourne un objet](#)
- ◆ [Prépare une transaction ibase](#)
- ◆ [Prépare à la lecture \(z39.50 present\)](#)
- ◆ [Puissance](#)
- ◆ [Puissance et modulo](#)
- **Q**
- ◆ [Quitte le canal](#)
- **R**
- ◆ [Racine carrée](#)
- ◆ [Racine carrée avec reste gmp](#)
- ◆ [Racine carrée gmp](#)
- ◆ [Rafraîchit l'écran](#)
- ◆ [Rafraîchit l'écran virtuel pour prendre en compte les relations entre les conteneurs de la pile](#)
- ◆ [Rafraîchit la fenêtre sur l'écran du terminal](#)
- ◆ [Ramène le pointeur interne d'un lob oracle au début](#)
- ◆ [Rapporte le message d'erreur du dernier appel de fonction dbx](#)
- ◆ [Rassemble des informations sur la langue et la configuration locale](#)
- ◆ [Rassemble les éléments d'un tableau en une chaîne](#)
- ◆ [Recherche 0](#)
- ◆ [Recherche 1](#)
- ◆ [Recherche dans un niveau](#)
- ◆ [Recherche dans un tableau la clé associée à une valeur](#)
- ◆ [Recherche des chemins qui vérifient un masque](#)
- ◆ [Recherche des objets](#)
- ◆ [Recherche la première occurrence dans une chaîne, sans tenir compte de la casse](#)
- ◆ [Recherche par expression rationnelle avec support des caractères multi-octets](#)
- ◆ [Recherche par expression rationnelle insensible à la casse](#)
- ◆ [Recherche par expression rationnelle multi-octets](#)
- ◆ [Recherche sur le serveur ldap](#)
- ◆ [Recherche tous les éléments avec un nom de balise donné dans un espace de noms spécifié](#)
- ◆ [Recherche un message dans le domaine courant](#)
- ◆ [Recherche un objet](#)
- ◆ [Recherche un objet](#)
- ◆ [Recherche un objet dans un groupe](#)
- ◆ [Recherche un objet dans un groupe](#)
- ◆ [Recherche un élément avec son identifiant](#)
- ◆ [Recherche un élément d'un type spécifique dans la base de données](#)
- ◆ [Recherche un événement dans le calendrier](#)
- ◆ [Recherche un événement dans le calendrier](#)
- ◆ [Recherche une chaîne de caractères dans un ensemble de caractères](#)
- ◆ [Rechercher et remplacer par expression rationnelle standard](#)
- ◆ [Rechercher/remplacer avec une expression rationnelle et fonction de callback](#)
- ◆ [Recodage de fichier à fichier, en fonction de la requête](#)
- ◆ [Recode une chaîne en fonction de la requête](#)
- ◆ [Reculé d'un cadre](#)
- ◆ [Reculé le pointeur courant de tableau](#)
- ◆ [Redonne les valeurs de configuration par défaut de tidy](#)
- ◆ [Redéfinit une constante déjà définie](#)

- ◆ [Redémarre l'interpréteur](#)
- ◆ [Referme le dictionnaire cracklib](#)
- ◆ [Referme une archive rar, et libère les ressources](#)
- ◆ [Rejoint un canal irc](#)
- ◆ [Relance la connexion au serveur postgresql](#)
- ◆ [Relie un mot](#)
- ◆ [Remet le pointeur interne de tableau au début](#)
- ◆ [Remet à sa position initiale le temporisateur d'apache](#)
- ◆ [Remet à zéro l'environnement graphique pdf](#)
- ◆ [Remet à zéro la session courante](#)
- ◆ [Remplace des caractères dans une chaîne](#)
- ◆ [Remplace des segments de chaînes, avec le support des expressions rationnelles mutli-octets](#)
- ◆ [Remplace l'identifiant de session courant par un nouveau](#)
- ◆ [Remplace la fenêtre associée à un conteneur](#)
- ◆ [Remplace le domaine courant](#)
- ◆ [Remplace le domaine lors d'une recherche](#)
- ◆ [Remplace le mot de passe d'un utilisateur virtuel](#)
- ◆ [Remplace les clés par les valeurs, et les valeurs par les clés](#)
- ◆ [Remplace les xincludes dans un objet domdocument](#)
- ◆ [Remplace ou insère une ligne dba](#)
- ◆ [Remplace toutes les occurrences dans une chaîne](#)
- ◆ [Remplace un attribut dans l'entrée courante](#)
- ◆ [Remplace un enregistrement dans une base dbase](#)
- ◆ [Remplace un fils](#)
- ◆ [Remplace un noeud domxml](#)
- ◆ [Remplace un noeud fils](#)
- ◆ [Remplace un objet](#)
- ◆ [Remplace un segment dans une chaîne](#)
- ◆ [Remplace une définition de fonction avec une nouvelle implémentation](#)
- ◆ [Remplace une sous-chaîne dans le noeud domcharacterdata node](#)
- ◆ [Remplace une valeur](#)
- ◆ [Remplace une valeur d'un élément existant](#)
- ◆ [Remplace xincludes dans un objet domdocument](#)
- ◆ [Remplacement par expression rationnelle](#)
- ◆ [Remplacement par expression rationnelle insensible à la casse](#)
- ◆ [Remplis un tableau avec une même valeur](#)
- ◆ [Remplit](#)
- ◆ [Remplit et passe le pinceau sur le chemin pdf courant](#)
- ◆ [Remplit et trace le chemin courant](#)
- ◆ [Remplit le chemin courant](#)
- ◆ [Remplit le chemin courant](#)
- ◆ [Remplit le chemin pdf courant avec la couleur courante](#)
- ◆ [Remplit le chemin, dessine le bord et ferme le chemin](#)
- ◆ [Remplit le chemin, et dessine le bord](#)
- ◆ [Remplit un champ de l'en-tête de document pdf](#)
- ◆ [Remplit un espace avec un ton](#)
- ◆ [Remplit une région avec une couleur spécifique](#)
- ◆ [Rend courant le contexte spécifié](#)
- ◆ [Renomme un calendrier](#)
- ◆ [Renomme un calendrier](#)
- ◆ [Renomme un fichier distant](#)
- ◆ [Renomme un fichier ou un dossier](#)
- ◆ [Renomme un fichier sur un serveur ftp](#)

- ◆ Renomme une boîte aux lettres
- ◆ Renomme une fonction intégrée dans la table des fonctions globales
- ◆ Renomme une relation
- ◆ Renvoie l'espace disque disponible dans le répertoire
- ◆ Renvoie la date de dernier accès à un inode
- ◆ Renvoie la date de dernière modification du fichier
- ◆ Renvoie la date à laquelle le fichier a été accédé pour la dernière fois
- ◆ Renvoie la ligne courante du fichier et élimine les balises html
- ◆ Renvoie la ligne courante et cherche les champs csv
- ◆ Renvoie la ligne courante sur laquelle se trouve le pointeur du fichier
- ◆ Renvoie la position du pointeur du fichier
- ◆ Renvoie la racine carrée d'un nombre de grande taille
- ◆ Renvoie la taille d'un fichier
- ◆ Renvoie le nom du dossier
- ◆ Renvoie le nom du fichier vers lequel pointe un lien symbolique
- ◆ Renvoie le nom du propriétaire du fichier
- ◆ Renvoie le numéro d'inode du fichier
- ◆ Renvoie les informations d'un lien
- ◆ Renvoie les informations à propos d'un fichier
- ◆ Renvoie les permissions affectées à un fichier
- ◆ Renvoie true: si ce node est jst
- ◆ Renvoie une chaîne en majuscules
- ◆ Renvoie une chaîne en minuscules
- ◆ Remplace l'itérateur au début
- ◆ Remplace le pointeur au début du fichier
- ◆ Remplace le pointeur de fichier au début
- ◆ Remplace le pointeur simplexml au début
- ◆ Remplace un caractère dans la queue d'entrée
- ◆ Reprend le téléchargement d'un fichier (non bloquant)
- ◆ Reprend un monitoring suspendu
- ◆ Représente un id global en un id virtuel local
- ◆ Repère la dernière occurrence d'un caractère dans une chaîne
- ◆ Repère la première occurrence d'un caractère dans une chaîne
- ◆ Repère un fichier à partir d'un masque
- ◆ Requiert les informations sur un utilisateur
- ◆ Restaure la position
- ◆ Restaure la valeur de l'option de configuration
- ◆ Restaure la valeur de la directive de configuration include_path
- ◆ Restaure le mode prog sauvé par def_prog_mode
- ◆ Restaure le mode shell, mode sauvé par def_shell_mode
- ◆ Restaure un contexte précédemment sauvegardé
- ◆ Restaure un environnement sauvegardé
- ◆ Restaure un gestionnaire d'url supprimé
- ◆ Restaure une sauvegarde de base de données interbase
- ◆ Reste de la division de deux nombres gmp
- ◆ Retarde l'affichage sur les terminaux utilisant des caractères de remplissage
- ◆ Retarde l'exécution en micro-secondes
- ◆ Retire un conteneur de la pile et l'efface (mais pas la fenêtre associée)
- ◆ Retire un conteneur de la pile, pour le rendre invisible
- ◆ Retire un curl multiple d'un jeu de curl

- ◆ [Retire une contrainte d'une relation](#)
- ◆ [Retourne true si ce noeud est en php](#)
- ◆ [Retourne true si ce noeud tidy est du code asp](#)
- ◆ [Retourne true si l'entrée courante est '.' ou '..'](#)
- ◆ [Retourne true si l'entrée est un fichier valide](#)
- ◆ [Retourne true si le code de retour représente une fin normale](#)
- ◆ [Retourne true si le code statut représente une fin due à un signal](#)
- ◆ [Retourne true si le fichier est accessible en lecture](#)
- ◆ [Retourne true si le fichier est exécutable](#)
- ◆ [Retourne true si le fichier est un dossier](#)
- ◆ [Retourne true si le fichier est un lien symbolique](#)
- ◆ [Retourne true si le fichier peut être modifié](#)
- ◆ [Retourne true si le processus fils est stoppé](#)
- ◆ [Retourne true si un objet a pour parent une classe donnée](#)
- ◆ [Retourne chaque paire clé/valeur d'un tableau](#)
- ◆ [Retourne des détails sur un calendrier](#)
- ◆ [Retourne des détails sur une colonne mysql](#)
- ◆ [Retourne des informations concernant les arguments d'une fonction](#)
- ◆ [Retourne des informations sur la configuration du serveur](#)
- ◆ [Retourne des informations sur la connexion courante](#)
- ◆ [Retourne des informations sur la file de messages](#)
- ◆ [Retourne des informations sur un chemin système](#)
- ◆ [Retourne des informations sur un fichier paradox](#)
- ◆ [Retourne des informations sur un groupe](#)
- ◆ [Retourne des informations sur un groupe](#)
- ◆ [Retourne des informations sur un utilisateur](#)
- ◆ [Retourne des informations sur un utilisateur](#)
- ◆ [Retourne des informations sur une colonne de résultat](#)
- ◆ [Retourne des informations à propos d'un paramètre dans une requête préparée ibase](#)
- ◆ [Retourne des informations à propos de la bibliothèque gd installée](#)
- ◆ [Retourne des informations à propos de la dernière requête exécutée](#)
- ◆ [Retourne des statistiques sur le serveur de bases](#)
- ◆ [Retourne des statistiques sur le serveur de cache de document](#)
- ◆ [Retourne des statistiques sur le serveur de textes](#)
- ◆ [Retourne des statistiques sur les caractères utilisés dans une chaîne](#)
- ◆ [Retourne l'adresse ip correspondant à un hôte](#)
- ◆ [Retourne l'arbre des messages organisés par thread](#)
- ◆ [Retourne l'attribut de la clé soft label courante](#)
- ◆ [Retourne l'en-tête d'un message](#)
- ◆ [Retourne l'encodage utilisé par la bibliothèque sqlite](#)
- ◆ [Retourne l'ensemble optimal de colonnes, qui permet de définir uniquement une ligne dans une table](#)
- ◆ [Retourne l'entrée courante de simplexml](#)
- ◆ [Retourne l'entrée courante du dossier](#)
- ◆ [Retourne l'entrée courante du tableau](#)
- ◆ [Retourne l'erreur rencontrée par la dernière opération ircg](#)
- ◆ [Retourne l'erreur sqlstate de la dernière opération mysql](#)
- ◆ [Retourne l'erreur sqlstate depuis la dernière opération maxdb](#)
- ◆ [Retourne l'extension du fichier pour le type d'image](#)
- ◆ [Retourne l'heure actuelle](#)
- ◆ [Retourne l'heure de coucher du soleil pour un jour et un endroit donnés](#)
- ◆ [Retourne l'heure de levé du soleil pour un jour et un endroit donnés](#)
- ◆ [Retourne l'id de l'utilisateur du processus courant](#)

- ◆ Retourne l'id du groupe de processus
- ◆ Retourne l'id effectif du groupe du processus courant
- ◆ Retourne l'identifiant automatiquement généré par la dernière requête
- ◆ Retourne l'identifiant de la dernière ligne
- ◆ Retourne l'identifiant de la dernière ligne insérée ou la valeur d'une séquence
- ◆ Retourne l'identifiant de la note non lue pour l'utilisateur user_name
- ◆ Retourne l'identifiant du groupe de processus
- ◆ Retourne l'identifiant du logo php
- ◆ Retourne l'identifiant du processus courant
- ◆ Retourne l'identifiant du processus parent
- ◆ Retourne l'identifiant du thread mysql courant
- ◆ Retourne l'identifiant du thread pour la connexion courante
- ◆ Retourne l'identifiant du thread pour la connexion courante
- ◆ Retourne l'identifiant généré par la dernière requête insert mysql
- ◆ Retourne l'identifiant maximal de hash
- ◆ Retourne l'identifiant public du type de document
- ◆ Retourne l'identifiant système du type de document
- ◆ Retourne l'identificateur demandé
- ◆ Retourne l'index d'une couleur avec son canal alpha
- ◆ Retourne l'index de l'octet courant d'un analyseur xml
- ◆ Retourne l'index de la couleur d'un pixel donné
- ◆ Retourne l'index de la couleur donnée
- ◆ Retourne l'index de la couleur donnée, ou la plus proche possible
- ◆ Retourne l'index de la couleur la plus proche d'une couleur donnée
- ◆ Retourne l'index de la ligne courante
- ◆ Retourne l'inode du script
- ◆ Retourne l'interlignage de la police courante à sa taille courante ou 0 s'il n'est pas disponible
- ◆ Retourne l'interlignage de la police ou 0 s'il n'est pas disponible
- ◆ Retourne l'objet propriétaire d'une ancre
- ◆ Retourne l'offset du début du prochain segment repéré par une expression rationnelle
- ◆ Retourne l'uid d'un message
- ◆ Retourne l'uid du groupe du processus courant
- ◆ Retourne l'uid du propriétaire du script actuel
- ◆ Retourne l'uid effectif de l'utilisateur du processus courant
- ◆ Retourne l'uri de l'espace de noms selon le préfixe
- ◆ Retourne l'url d'une animation shockwave flash
- ◆ Retourne l'échelle d'un champ
- ◆ Retourne l'élément courant du tableau
- ◆ Retourne l'équivalent de l'appel système select() sur un tableau de flux avec un délai d'expiration spécifié par tv_sec et tv_usec
- ◆ Retourne l'état de bufferisation lob d'oracle
- ◆ Retourne l'état de la dernière requête sesam
- ◆ Retourne l'état du service défini par serviceid
- ◆ Retourne la chaîne correspondant à l'erreur donnée
- ◆ Retourne la cible d'un lien symbolique
- ◆ Retourne la cible du noeud de l'instruction en cours
- ◆ Retourne la classe d'un objet
- ◆ Retourne la classe parente d'une classe
- ◆ Retourne la clé courante du tableau
- ◆ Retourne la clé courante simplexml
- ◆ Retourne la clé d'un attribut
- ◆ Retourne la configuration actuelle de l'option magic_quotes_gpc

- ◆ Retourne la configuration actuelle de l'option `magic_quotes_runtime`
- ◆ Retourne la configuration actuelle du `cache_expire`
- ◆ Retourne la couleur associée à un index
- ◆ Retourne la couleur la plus proche, en tenant compte du canal alpha
- ◆ Retourne la date de dernière modification d'un fichier sur un serveur ftp
- ◆ Retourne la date de dernière modification de la page
- ◆ Retourne la date de publication (version) de la bibliothèque tidy
- ◆ Retourne la date/heure
- ◆ Retourne la dernière erreur (si elle existe) qui est survenue lors de la dernière requête
- ◆ Retourne la dernière erreur oracle
- ◆ Retourne la dernière note du serveur postgresql
- ◆ Retourne la dernière requête soap
- ◆ Retourne la dernière réponse soap
- ◆ Retourne la description d'une raison
- ◆ Retourne la description du terminal
- ◆ Retourne la destination d'une ancre source
- ◆ Retourne la documentation pour le nom de l'option donnée
- ◆ Retourne la fenêtre associée avec un conteneur
- ◆ Retourne la forme glyphe d'un caractère en tant que chaîne de caractères
- ◆ Retourne la hampe de la police courante à sa taille courante ou 0 si elle n'est pas disponible
- ◆ Retourne la hampe de la police ou 0 si elle est non disponible
- ◆ Retourne la hauteur d'une bitmap flash
- ◆ Retourne la hauteur de l'image
- ◆ Retourne la hauteur de la police
- ◆ Retourne la hauteur du a majuscule, et du x minuscule
- ◆ Retourne la largeur d'un texte avec la police courante
- ◆ Retourne la largeur d'une bitmap flash
- ◆ Retourne la largeur d'une chaîne
- ◆ Retourne la largeur d'une image
- ◆ Retourne la largeur de la police
- ◆ Retourne la ligne associée
- ◆ Retourne la ligne courante d'un jeu de résultat sous forme d'objet
- ◆ Retourne la ligne courante d'un jeu de résultats dans un objet
- ◆ Retourne la ligne du caractère actuellement supprimé
- ◆ Retourne la ligne suivante d'un résultat oracle dans un tableau
- ◆ Retourne la ligne suivante du jeu de résultats en tant qu'objet
- ◆ Retourne la liste d'ip correspondante à un hôte
- ◆ Retourne la liste de tous les modules compilés et chargés
- ◆ Retourne la liste de toutes les ancres
- ◆ Retourne la liste des attributs
- ◆ Retourne la liste des constantes et leurs valeurs
- ◆ Retourne la liste des en-têtes de réponse du script courant
- ◆ Retourne la liste des entités
- ◆ Retourne la liste des fichiers .ini analysés dans les dossiers de configuration supplémentaires
- ◆ Retourne la liste des fichiers dans un dossier
- ◆ Retourne la liste des fonctions définies
- ◆ Retourne la liste des imprimantes attachées à un serveur
- ◆ Retourne la liste des modules apache chargés
- ◆ Retourne la liste des notations
- ◆ Retourne la liste des utilisateurs connectés
- ◆ Retourne la longueur d'une chaîne

- ◆ Retourne la longueur des colonnes de la ligne courante dans un jeu de résultats
- ◆ Retourne la longueur des colonnes de la ligne courante du jeu de résultats
- ◆ Retourne la longueur du contenu du buffer de sortie
- ◆ Retourne la méthode de compression utilisée avec gzip
- ◆ Retourne la méthode avec laquelle les valeurs snmp seront retournées
- ◆ Retourne la partie entière d'un variant
- ◆ Retourne la pile d'appel courante dans un tableau
- ◆ Retourne la portion entière d'un variant
- ◆ Retourne la position courante
- ◆ Retourne la position courante dans un objet de grande taille postgresql
- ◆ Retourne la position courante du curseur pour une fenêtre
- ◆ Retourne la position courante du pointeur de lob
- ◆ Retourne la position du champ courant dans un pointeur de résultat
- ◆ Retourne la position et la longueur du segment de chaîne qui vérifie le masque de l'expression rationnelle
- ◆ Retourne la première entrée
- ◆ Retourne la première raison d'erreur
- ◆ Retourne la première référence
- ◆ Retourne la priorité d'un processus
- ◆ Retourne la prochaine occurrence d'un événement
- ◆ Retourne la précision d'un champ
- ◆ Retourne la représentation de l'élément courant sous forme de chaîne
- ◆ Retourne la somme de contrôle crc32 d'une chaîne
- ◆ Retourne la source sans commentaires ni espaces blancs
- ◆ Retourne la table de traduction des entités utilisée par htmlspecialchars et htmlentities
- ◆ Retourne la taille d'un blob
- ◆ Retourne la taille d'un blob ibase et d'autres informations utiles
- ◆ Retourne la taille d'un champ
- ◆ Retourne la taille d'un champ
- ◆ Retourne la taille d'un champ de résultat mysql
- ◆ Retourne la taille d'un champ filepro
- ◆ Retourne la taille d'un champ oracle
- ◆ Retourne la taille d'un clob
- ◆ Retourne la taille d'un dossier
- ◆ Retourne la taille d'un fichier
- ◆ Retourne la taille d'un lob oracle
- ◆ Retourne la taille d'une chaîne
- ◆ Retourne la taille d'une collection oracle
- ◆ Retourne la taille d'une colonne ovrmos
- ◆ Retourne la taille d'une fenêtre
- ◆ Retourne la taille d'une image
- ◆ Retourne la taille de bloc du hash
- ◆ Retourne la taille de blocs d'un algorithme
- ◆ Retourne la taille de blocs d'un algorithme
- ◆ Retourne la taille de blocs d'un chiffrement
- ◆ Retourne la taille de chaque colonne d'une ligne de résultat mysql
- ◆ Retourne la taille de la chaîne
- ◆ Retourne la taille de la clé d'un chiffrement
- ◆ Retourne la taille de la police courante
- ◆ Retourne la taille du vi d'un algorithme
- ◆ Retourne la taille du vi utilisé par un couple chiffrement/mode
- ◆ Retourne la taille imprimée
- ◆ Retourne la taille interne de stockage d'un champ donné

- ◆ Retourne la taille maximale de clé
- ◆ Retourne la taille maximale de la clé pour un mode
- ◆ Retourne la valeur absolue d'un variant
- ◆ Retourne la valeur d'un attribut
- ◆ Retourne la valeur d'un attribut dans une langue
- ◆ Retourne la valeur d'un champ fdf
- ◆ Retourne la valeur d'un champ filepro
- ◆ Retourne la valeur d'un noeud
- ◆ Retourne la valeur d'un élément d'une collection oracle
- ◆ Retourne la valeur d'une colonne dans un résultat oracle
- ◆ Retourne la valeur d'une constante
- ◆ Retourne la valeur d'une option de connexion yaz
- ◆ Retourne la valeur d'une option de php
- ◆ Retourne la valeur d'une variable d'environnement
- ◆ Retourne la valeur de l'attribut
- ◆ Retourne la valeur de l'attribut
- ◆ Retourne la valeur de l'attribut spécifié
- ◆ Retourne la valeur de l'option de configuration tidy
- ◆ Retourne la valeur de la variable, au format chaîne
- ◆ Retourne la valeur de pi
- ◆ Retourne la valeur numérique entière équivalent d'une variable
- ◆ Retourne la valeur rgb d'une couleur
- ◆ Retourne la valeur textuelle du dernier appel à une fonction
- ◆ Retourne la valeur à l'\$index spécifié
- ◆ Retourne la version courante de curl
- ◆ Retourne la version de la bibliothèque sqlite
- ◆ Retourne la version du client maxdb sous la forme d'une chaîne
- ◆ Retourne la version du protocole maxdb utilisé
- ◆ Retourne la version du protocole mysql utilisé
- ◆ Retourne la version du serveur maxdb
- ◆ Retourne la version du serveur maxdb sous la forme d'un entier
- ◆ Retourne la version du serveur mysql
- ◆ Retourne le acl pour la boîte aux lettres
- ◆ Retourne le baudrate du terminal
- ◆ Retourne le chemin canonique absolu
- ◆ Retourne le chemin du dossier
- ◆ Retourne le chemin du terminal
- ◆ Retourne le chemin et le nom de l'entrée courante du dossier
- ◆ Retourne le chemin et le nom de l'entrée courante du dossier
- ◆ Retourne le code ascii d'un caractère
- ◆ Retourne le code d'erreur
- ◆ Retourne le code d'erreur bzip2
- ◆ Retourne le code d'erreur de la connexion mysql
- ◆ Retourne le code d'erreur de la dernière opération fdf
- ◆ Retourne le code d'erreur de la dernière opération nis
- ◆ Retourne le code d'erreur de la dernière requête informix
- ◆ Retourne le code d'erreur frontbase
- ◆ Retourne le code d'erreur ibase
- ◆ Retourne le code d'erreur oracle
- ◆ Retourne le code d'un processus fils terminé

- ◆ Retourne le code erreur du dernier appel à la connexion
- ◆ Retourne le code erreur pour le dernier appel à une fonction
- ◆ Retourne le code erreur pour le dernier appel à une fonction de requête
- ◆ Retourne le code php utilisé pour générer une variable
- ◆ Retourne le code sqlstate de la dernière opération mysql
- ◆ Retourne le conteneur au-dessus du conteneur
- ◆ Retourne le conteneur sous le conteneur
- ◆ Retourne le contenu d'un objet
- ◆ Retourne le contenu d'un objet blob
- ◆ Retourne le contenu d'un objet char
- ◆ Retourne le contenu de la variable sqlca.sqlerrd[0..5] après une requête
- ◆ Retourne le contenu du buffer de sortie
- ◆ Retourne le contenu du noeud
- ◆ Retourne le contenu obtenu avec l'option curlopt_returntransfer
- ◆ Retourne le couple (clé : valeur) suivant d'une carte donnée
- ◆ Retourne le dernier code d'erreur sqlite
- ◆ Retourne le dernier fils du noeud
- ◆ Retourne le dernier identifiant généré automatiquement par la dernière requête
- ◆ Retourne le dernier message d'erreur du serveur
- ◆ Retourne le dernier message du serveur
- ◆ Retourne le document auquel appartient ce noeud dom xml
- ◆ Retourne le domaine nis par défaut
- ◆ Retourne le dossier de travail courant
- ◆ Retourne le drapeau d'action pour keypress(char)
- ◆ Retourne le fichier courant ou la position courante
- ◆ Retourne le fils d'un objet
- ◆ Retourne le fils de l'itérateur interne comme un cachingrecursiveiterator
- ◆ Retourne le fils de l'itérateur interne contenu dans parentiterator
- ◆ Retourne le frère précédent d'un noeud
- ◆ Retourne le frère précédent de ce noeud
- ◆ Retourne le gid du propriétaire du script
- ◆ Retourne le jambage de la police courante à sa taille courante ou 0 s'il n'est pas disponible
- ◆ Retourne le jambage de la police ou 0 s'il n'est pas disponible
- ◆ Retourne le jeu de caractères courant pour la connexion
- ◆ Retourne le jeu de caractères courant pour les expressions rationnelles
- ◆ Retourne le jeu de caractères par défaut pour la connexion à la base de données
- ◆ Retourne le logo de zend
- ◆ Retourne le message d'erreur bzip2
- ◆ Retourne le message d'erreur de connexion mysql
- ◆ Retourne le message d'erreur de la dernière opération qdom. ou false si aucune erreur n'est survenue
- ◆ Retourne le message d'erreur de la dernière requête informix
- ◆ Retourne le message d'erreur fdf
- ◆ Retourne le message d'erreur frontbase
- ◆ Retourne le message d'erreur msql
- ◆ Retourne le message d'erreur openssl
- ◆ Retourne le message d'erreur oracle
- ◆ Retourne le message d'erreur sesam
- ◆ Retourne le message d'erreur sqlite
- ◆ Retourne le message de diagnostic
- ◆ Retourne le message ldap de la dernière commande ldap
- ◆ Retourne le niveau d'utilisation des ressources
- ◆ Retourne le noeud d'attribut

- ◆ Retourne le noeud d'attribut
- ◆ Retourne le noeud frère suivant
- ◆ Retourne le noeud racine d'un document dom xml
- ◆ Retourne le nom court du terminal
- ◆ Retourne le nom d'hôte
- ◆ Retourne le nom d'hôte correspondant à une ip
- ◆ Retourne le nom d'un champ dans le résultat d'une requête
- ◆ Retourne le nom d'un champ filepro
- ◆ Retourne le nom d'un champ postgresql
- ◆ Retourne le nom d'une colonne
- ◆ Retourne le nom d'une colonne dans un résultat mysql
- ◆ Retourne le nom d'une colonne oracle
- ◆ Retourne le nom d'une colonne ovrimos
- ◆ Retourne le nom d'utilisateur pour la connexion
- ◆ Retourne le nom de device du terminal
- ◆ Retourne le nom de l'algorithme
- ◆ Retourne le nom de l'attribut
- ◆ Retourne le nom de l'entrée courante du dossier
- ◆ Retourne le nom de l'élément courant
- ◆ Retourne le nom de la base de données postgresql
- ◆ Retourne le nom de la classe d'un objet
- ◆ Retourne le nom de la colonne de résultat oracle
- ◆ Retourne le nom de la machine maître pour une carte
- ◆ Retourne le nom de la table mysql où se trouve une colonne
- ◆ Retourne le nom de login
- ◆ Retourne le nom de protocole associé à un numéro de protocole
- ◆ Retourne le nom de tty associé à la connexion
- ◆ Retourne le nom du backend
- ◆ Retourne le nom du champ sqlite
- ◆ Retourne le nom du curseur ovrimos
- ◆ Retourne le nom du dossier courant
- ◆ Retourne le nom du hash
- ◆ Retourne le nom du jeu de caractères utilisé par le client mysql
- ◆ Retourne le nom du mode
- ◆ Retourne le nom du mois
- ◆ Retourne le nom du noeud
- ◆ Retourne le nom du possesseur du script courant
- ◆ Retourne le nom du système
- ◆ Retourne le nom du type de document
- ◆ Retourne le nombre courant de colonne d'un analyseur xml
- ◆ Retourne le nombre d'alertes d'accessibilité tidy rencontrées dans le document
- ◆ Retourne le nombre d'alertes tidy rencontrées dans le document spécifié
- ◆ Retourne le nombre d'arguments passés à la fonction
- ◆ Retourne le nombre d'attributs
- ◆ Retourne le nombre d'avertissements depuis la dernière requête pour un lien donné
- ◆ Retourne le nombre d'avertissements générés par la dernière requête
- ◆ Retourne le nombre d'enregistrements d'une base de données paradox
- ◆ Retourne le nombre d'erreurs de configuration tidy rencontrées dans le document
- ◆ Retourne le nombre d'erreurs tidy rencontrées dans le document
- ◆ Retourne le nombre d'éléments dans l'itérateur
- ◆ Retourne le nombre de caractères d'une chaîne
- ◆ Retourne le nombre de champ
- ◆ Retourne le nombre de champs d'un résultat mysql
- ◆ Retourne le nombre de champs d'une base de données paradox

- ◆ Retourne le nombre de champs dans un résultat
- ◆ Retourne le nombre de champs dans un résultat
- ◆ Retourne le nombre de champs dans un résultat ms sql server
- ◆ Retourne le nombre de champs dans un résultat sqlite
- ◆ Retourne le nombre de champs dans un résultat sybase
- ◆ Retourne le nombre de champs dans une base filepro
- ◆ Retourne le nombre de champs renvoyés par la dernière requête
- ◆ Retourne le nombre de colonnes
- ◆ Retourne le nombre de colonnes pour la dernière requête
- ◆ Retourne le nombre de colonnes dans le jeu de résultats
- ◆ Retourne le nombre de colonnes dans un résultat
- ◆ Retourne le nombre de colonnes dans un résultat ibase
- ◆ Retourne le nombre de colonnes dans un résultat oracle
- ◆ Retourne le nombre de colonnes dans une requête
- ◆ Retourne le nombre de colonnes ovrimos
- ◆ Retourne le nombre de colonnes pour la dernière requête
- ◆ Retourne le nombre de jours dans un mois
- ◆ Retourne le nombre de jours dans un mois, pour une année et un calendrier donné
- ◆ Retourne le nombre de jours entre le 21 mars et pâques, pour une année donnée
- ◆ Retourne le nombre de lignes
- ◆ Retourne le nombre de lignes affectées
- ◆ Retourne le nombre de lignes affectées
- ◆ Retourne le nombre de lignes affectées durant la dernière commande oracle
- ◆ Retourne le nombre de lignes affectées lors de la dernière opération mysql
- ◆ Retourne le nombre de lignes affectées ou renvoyées par la dernière requête
- ◆ Retourne le nombre de lignes affectées par la dernière opération mysql
- ◆ Retourne le nombre de lignes affectées par la dernière requête ibase
- ◆ Retourne le nombre de lignes affectées par la dernière requête sybase
- ◆ Retourne le nombre de lignes affectées par le dernier appel à la fonction `pdostatement::execute()`
- ◆ Retourne le nombre de lignes affectées par une modification ovrimos
- ◆ Retourne le nombre de lignes affectées par une requête
- ◆ Retourne le nombre de lignes affectées par une requête ms sql server
- ◆ Retourne le nombre de lignes d'un résultat mysql
- ◆ Retourne le nombre de lignes d'un résultat mysql
- ◆ Retourne le nombre de lignes d'un résultat sqlite
- ◆ Retourne le nombre de lignes dans le jeu de résultats
- ◆ Retourne le nombre de lignes dans un résultat
- ◆ Retourne le nombre de lignes dans un résultat
- ◆ Retourne le nombre de lignes dans un résultat ms sql
- ◆ Retourne le nombre de lignes dans un résultat sybase
- ◆ Retourne le nombre de lignes dans une base filepro
- ◆ Retourne le nombre de lignes postgresql
- ◆ Retourne le nombre de lignes qui ont été modifiées par la dernière requête sqlite
- ◆ Retourne le nombre de marqueurs pour une requête donnée
- ◆ Retourne le nombre de message dans la boîte aux lettres courante
- ◆ Retourne le nombre de messages récents dans la boîte aux lettres courante
- ◆ Retourne le nombre de niveaux d'imbrications du système de bufferisation de sortie
- ◆ Retourne le nombre de paramètres d'une commande mysql
- ◆ Retourne le nombre de paramètres dans une requête préparée ibase
- ◆ Retourne le nombre de raisons d'erreurs
- ◆ Retourne le nombre de résultats de la dernière recherche
- ◆ Retourne le nombre maximal de valeurs d'une collection oracle
- ◆ Retourne le nombre total de lignes modifiées, effacées ou insérées par la dernière

- requête
- ◆ Retourne le nombre total de lignes modifiées, effacées ou insérées par la dernière requête
- ◆ Retourne le numéro d'erreur
- ◆ Retourne le numéro d'erreur xslt courant
- ◆ Retourne le numéro d'erreur de la dernière commande mysql
- ◆ Retourne le numéro d'erreur ldap de la dernière commande exécutée
- ◆ Retourne le numéro d'instance associé avec le path
- ◆ Retourne le numéro d'instance associé à un comment
- ◆ Retourne le numéro d'ordre d'une carte
- ◆ Retourne le numéro d'une colonne
- ◆ Retourne le numéro de cadre courant
- ◆ Retourne le numéro de colonne ovrimos
- ◆ Retourne le numéro de la version courante de php
- ◆ Retourne le numéro de ligne courant d'un analyseur xml
- ◆ Retourne le numéro de ligne de la dernière ligne insérée
- ◆ Retourne le numéro de port
- ◆ Retourne le numéro de port associé à un service internet et un protocole
- ◆ Retourne le numéro de processus courant de php
- ◆ Retourne le numéro de protocole associé à un nom de protocole
- ◆ Retourne le numéro de semaine dans l'année pour une date donnée
- ◆ Retourne le numéro de séquence de message pour un uid donné
- ◆ Retourne le numéro de version de payflow pro
- ◆ Retourne le numéro de version de sablotron
- ◆ Retourne le numéro de version du serveur
- ◆ Retourne le numéro de version majeur de pdflib
- ◆ Retourne le numéro de version mineure de pdflib
- ◆ Retourne le numéro du jour de la semaine
- ◆ Retourne le numéro et le message d'erreur bzip2 dans un tableau
- ◆ Retourne le parent d'un objet
- ◆ Retourne le premier attribut
- ◆ Retourne le premier couple (clé : valeur) d'une carte donnée
- ◆ Retourne le premier fils du noeud
- ◆ Retourne le prochain champ dans le jeu de résultats
- ◆ Retourne le prochain champs d'un jeu de résultats
- ◆ Retourne le prochain frère du noeud courant
- ◆ Retourne le prochain identifiant d'objet libre
- ◆ Retourne le prochain lot de lignes ms sql server
- ◆ Retourne le prochain événement fam en attente
- ◆ Retourne le propriétaire d'un objet
- ◆ Retourne le préfixe d'espace de nom d'un noeud
- ◆ Retourne le préfixe de l'espace de noms selon l'uri de l'espace de noms
- ◆ Retourne le père d'un noeud dom xml
- ◆ Retourne le rectangle entourant un texte et dessiné avec une police postscript type1
- ◆ Retourne le rectangle entourant un texte et dessiné avec une police truetype
- ◆ Retourne le reste d'une division entre nombres de grande taille
- ◆ Retourne le reste de la division
- ◆ Retourne le résultat d'un scan
- ◆ Retourne le résultat de exp(number) - 1, calculé de manière précise, même si val est proche de 0
- ◆ Retourne le résultat de la division de deux variants
- ◆ Retourne le résultat de la fonction puissance avec deux variants
- ◆ Retourne le résultat de log(1 + number), calculé de manière précise, même si val est proche de zéro

- ◆ Retourne le schéma de la base de données
- ◆ Retourne le secret partagé
- ◆ Retourne le segment de chaîne trouvé par une expression rationnelle multi-octets
- ◆ Retourne le service internet qui correspond au port et protocole
- ◆ Retourne le sid du processus
- ◆ Retourne le signal qui a causé l'arrêt du processus fils
- ◆ Retourne le signal qui a provoqué la fin du processus fils
- ◆ Retourne le sous ensemble interne (internal subset)
- ◆ Retourne le sqlstate associé avec la dernière opération sur la base de données
- ◆ Retourne le status du processus de communication
- ◆ Retourne le statut courant du serveur mysql
- ◆ Retourne le statut de la chaîne
- ◆ Retourne le statut de la transaction en cours du serveur
- ◆ Retourne le statut du document spécifié
- ◆ Retourne le texte associé avec l'erreur générée lors de la dernière requête
- ◆ Retourne le timestamp unix actuel
- ◆ Retourne le timestamp unix actuel avec microsecondes
- ◆ Retourne le timestamp unix d'une date
- ◆ Retourne le timestamp unix d'une date gmt
- ◆ Retourne le titre de l'attribut
- ◆ Retourne le type d'interface utilisée entre le serveur web et php
- ◆ Retourne le type d'un champ dans le résultat d'une requête
- ◆ Retourne le type d'un champ filepro
- ◆ Retourne le type d'un champ postgresql donné par index
- ◆ Retourne le type d'un objet variant
- ◆ Retourne le type d'une colonne mysql spécifique
- ◆ Retourne le type de commande oracle
- ◆ Retourne le type de document
- ◆ Retourne le type de données d'un champ oracle
- ◆ Retourne le type de fichier
- ◆ Retourne le type de la colonne de résultat oracle
- ◆ Retourne le type de la variable
- ◆ Retourne le type de noeud
- ◆ Retourne le type de raison
- ◆ Retourne le type de requête rpl
- ◆ Retourne le type de ressource
- ◆ Retourne le type id (oid) pour le numéro du champ correspondant
- ◆ Retourne le type mime
- ◆ Retourne le type numérique d'une colonne ovrimos
- ◆ Retourne le type xmlrpc d'une valeur php
- ◆ Retourne les alertes et erreurs qui sont survenues lors de l'analyse du document
- ◆ Retourne les ancrages qui pointent sur un objet
- ◆ Retourne les arguments d'une fonction sous forme de tableau
- ◆ Retourne les attributs et verrouille l'objet
- ◆ Retourne les bits de statut de la connexion http
- ◆ Retourne les classes spl disponibles
- ◆ Retourne les données de résultat
- ◆ Retourne les données du noeud l'instruction en cours
- ◆ Retourne les données enregistrées dans une colonne mysql sous forme d'objet
- ◆ Retourne les en-têtes de la dernière requête soap
- ◆ Retourne les en-têtes de la dernière réponse soap
- ◆ Retourne les en-têtes de tous les messages d'une boîte aux lettres
- ◆ Retourne les enregistrements mx d'un hôte
- ◆ Retourne les fils d'un document distant

- ◆ Retourne les fils d'un noeud
- ◆ Retourne les identifiants du groupe du processus courant
- ◆ Retourne les informations associées à l'erreur lors de la dernière opération sur la base de données
- ◆ Retourne les informations de statut sur une boîte aux lettres autres que la boîte courante
- ◆ Retourne les informations sur le système d'exploitation
- ◆ Retourne les informations sur les paramètres de compilation du backend
- ◆ Retourne les informations sur n fichier ou un lien symbolique
- ◆ Retourne les interfaces implémentées dans une classe donnée
- ◆ Retourne les limites système
- ◆ Retourne les méta-données de préparation de requête mysql
- ◆ Retourne les méta-données pour une colonne d'un jeu de résultats
- ◆ Retourne les noms des méthodes d'une classe
- ◆ Retourne les notes rangées dans database_name
- ◆ Retourne les options postgresql
- ◆ Retourne les sockets associées à curl, pour utilisation
- ◆ Retourne les sources d'un objet de destination
- ◆ Retourne les spécifications d'un champ
- ◆ Retourne les statistiques sur le serveur hyperwave
- ◆ Retourne les types d'images supportés par la version courante de php
- ◆ Retourne les valeurs d'un résultat
- ◆ Retourne les valeurs d'un tableau
- ◆ Retourne les valeurs par défaut des propriétés d'une classe
- ◆ Retourne les éléments par leur nom de balise
- ◆ Retourne plus de détails après une erreur
- ◆ Retourne si \$index existe
- ◆ Retourne si oui ou non il reste des lignes disponibles
- ◆ Retourne si oui ou non une ligne précédente est disponible
- ◆ Retourne this (requis pour l'interface iterator)
- ◆ Retourne tous les en-têtes envoyés par le serveur en réponse à une requête http
- ◆ Retourne tous les objets y compris leur identifiant d'objet dans celui spécifié
- ◆ Retourne tous les objets, y compris leur id d'objet
- ◆ Retourne tout ou partie d'un résultat sesam
- ◆ Retourne toutes les alertes
- ◆ Retourne toutes les clés d'un tableau
- ◆ Retourne toutes les erreurs
- ◆ Retourne toutes les lignes d'un jeu de résultats en tant que tableau de tableaux
- ◆ Retourne toutes les options supportées par le terminal, combinées ensemble par l'opérateur or
- ◆ Retourne toutes les ressources persistantes dans un tableau
- ◆ Retourne toutes les ressources régulières courantes dans un tableau
- ◆ Retourne toutes les valeurs de l'attribut
- ◆ Retourne toutes les valeurs qui sont des énumérations avec leur valeur d'énumération au lieu de l'entier
- ◆ Retourne true si le noeud a des enfants
- ◆ Retourne true si le noeud a des frères
- ◆ Retourne true si le noeud est une partie d'un document html
- ◆ Retourne true si le noeud représente du texte (aucun marquage)
- ◆ Retourne true si le noeud représente un commentaire
- ◆ Retourne un analyseur destiné au traitement d'un message
- ◆ Retourne un caractère spécifique
- ◆ Retourne un champ d'un résultat mysql
- ◆ Retourne un champ individuel d'un rapport d'erreur

- ◆ Retourne un code d'erreur pour la dernière commande mysql
- ◆ Retourne un code erreur pour la dernière requête
- ◆ Retourne un document
- ◆ Retourne un document distant
- ◆ Retourne un document fdf sous forme de chaîne
- ◆ Retourne un document texte
- ◆ Retourne un document texte
- ◆ Retourne un enregistrement d'une base de données paradox
- ◆ Retourne un entier représentant la version du serveur mysql
- ◆ Retourne un entier représentant le dernier message d'erreur curl
- ◆ Retourne un identifiant de note, dans le serveur database_name
- ◆ Retourne un identifiant de sémaphore
- ◆ Retourne un identifiant de type de serveur ftp
- ◆ Retourne un identifiant unique pour une relation
- ◆ Retourne un index de couleur ou son alternative la plus proche, y compris le canal alpha
- ◆ Retourne un itérateur pour l'entrée courante si c'est un dossier
- ◆ Retourne un itérateur pour l'entrée courante si c'est un répertoire
- ◆ Retourne un itérateur pour l'entrée courante, si c'est un objet simplexml
- ◆ Retourne un message d'erreur
- ◆ Retourne un noeud selon son index
- ◆ Retourne un noeud spécifié par nom nom
- ◆ Retourne un noeud spécifié par son index
- ◆ Retourne un noeud spécifié par son nom local et son espace de nom
- ◆ Retourne un nouveau curl multiple
- ◆ Retourne un nouvel objet swfsound d'un fichier donné
- ◆ Retourne un objet contenant la structure de date
- ◆ Retourne un objet de classe hw_api
- ◆ Retourne un objet de compartiment depuis le corps pour des opérations sur celui-ci
- ◆ Retourne un objet représentant l'instance actuelle d'un objet com
- ◆ Retourne un objet snmp
- ◆ Retourne un objet swfprebuiltclip
- ◆ Retourne un objet swfvideostream
- ◆ Retourne un objet tidynode commençant à la balise <html>
- ◆ Retourne un objet tidynode représentant la racine du document html
- ◆ Retourne un objet tidynode à partir de la balise <head>
- ◆ Retourne un objet tidynode, commencé à partir de la balise <body>
- ◆ Retourne un résultat
- ◆ Retourne un segment de chaîne
- ◆ Retourne un tableau associatif avec des informations sur le message
- ◆ Retourne un tableau associatif avec les méthodes et propriétés de chaque classe qui a été agrégée
- ◆ Retourne un tableau associatif des propriétés d'un objet
- ◆ Retourne un tableau associatif pour la ligne courant de résultat ms sql server
- ◆ Retourne un tableau avec les différentes sections mime du message
- ◆ Retourne un tableau avec les noeuds placés sous le noeud courant, avec l'identifiant spécifié
- ◆ Retourne un tableau avec les noms des fichiers qui sont inclus dans un script
- ◆ Retourne un tableau avec les résultats de la recherche
- ◆ Retourne un tableau avec les versions du client, du protocole et du serveur (si disponible)

- ◆ Retourne un tableau avec noeuds pour le nom de tag donné dans le document ou un tableau vide si non trouvé
- ◆ Retourne un tableau avec toutes les interfaces déclarées
- ◆ Retourne un tableau contenant les tailles de clés acceptées par un algorithme
- ◆ Retourne un tableau contenant les tailles de clés supportées par un algorithme de chiffrement
- ◆ Retourne un tableau contenant tous les encodages supportés
- ◆ Retourne un tableau contenant toutes les lignes du jeu d'enregistrements
- ◆ Retourne un tableau d'objets représentant les champs dans le résultat
- ◆ Retourne un tableau de chaînes de caractères qui représente les clés qui peuvent être utilisées en tant que paramètres de réponse sur cette transaction
- ◆ Retourne un tableau de message après recherche
- ◆ Retourne un tableau de ressource représentant les champs dans un jeu de résultats
- ◆ Retourne un tableau des pilotes pdo disponibles
- ◆ Retourne un tableau des types de colonnes d'une certaine table
- ◆ Retourne un tableau indexé numériquement des variables superglobales enregistrées
- ◆ Retourne un timestamp unix pour pâques, à minuit
- ◆ Retourne un élément de la liste des arguments
- ◆ Retourne une adresse email formatée correctement
- ◆ Retourne une chaîne contenant la version du client mysql
- ◆ Retourne une chaîne contenant le dernier message d'erreur curl
- ◆ Retourne une chaîne contenant le type de connexion utilisée
- ◆ Retourne une chaîne contenant les informations de version du serveur oracle
- ◆ Retourne une chaîne descriptive de la dernière erreur
- ◆ Retourne une chaîne décrivant la dernière erreur
- ◆ Retourne une chaîne décrivant un message d'erreur
- ◆ Retourne une chaîne formatée
- ◆ Retourne une chaîne formatée
- ◆ Retourne une chaîne représentant la dernière erreur
- ◆ Retourne une chaîne représentant le typde de connexion utilisé
- ◆ Retourne une chaîne représentant les balises telles qu'analysées par tidy
- ◆ Retourne une chaîne xml basée sur un élément simplexml
- ◆ Retourne une clé d'un tableau associatif
- ◆ Retourne une colonne depuis la ligne suivante d'un jeu de résultats
- ◆ Retourne une description de l'erreur
- ◆ Retourne une description de la dernière erreur
- ◆ Retourne une description de la dernière erreur de traitement
- ◆ Retourne une donnée d'une ligne de résultat oracle
- ◆ Retourne une erreur sqlstate depuis la requête précédente
- ◆ Retourne une ligne de résultat
- ◆ Retourne une ligne de résultat ms sql server sous la forme d'un objet
- ◆ Retourne une ligne de résultat mysql sous la forme d'un objet
- ◆ Retourne une ligne de résultat mysql sous la forme d'un tableau
- ◆ Retourne une ligne de résultat mysql sous la forme d'un tableau associatif, d'un tableau indexé, ou les deux
- ◆ Retourne une ligne de résultat sous la forme d'un tableau associatif, d'un tableau indexé, ou les deux
- ◆ Retourne une ligne sous la forme d'un tableau
- ◆ Retourne une ligne sous la forme d'un tableau énuméré
- ◆ Retourne une ligne sybase sous la forme d'un objet
- ◆ Retourne une ligne sybase sous la forme d'un tableau
- ◆ Retourne une ligne sybase sous la forme d'un tableau numérique
- ◆ Retourne une liste d'événements entre deux dates

- ◆ Retourne une liste d'événements pour une date ou entre deux dates
- ◆ Retourne une liste d'événements qui ont une alarme prévue à une date
- ◆ Retourne une liste d'événements qui ont une alarme prévue à une date
- ◆ Retourne une liste de fonctions soap
- ◆ Retourne une liste de méthodes négociées
- ◆ Retourne une liste des types soap
- ◆ Retourne une partie de chaîne iconv
- ◆ Retourne une représentation date en variant d'un timestamp unix
- ◆ Retourne une requête de type rpl
- ◆ Retourne une ressource sur une section d'un message mime
- ◆ Retourne une section extraite du corps d'un message
- ◆ Retourne une somme de contrôle hash32
- ◆ Retourne une tableau contenant la carte entière
- ◆ Retourne à la première entrée du dossier
- ◆ Retourne les méta-données du jeu de résultats pour une requête préparée
- ◆ Returns large object's contents
- ◆ Revient au début du dossier
- ◆ Revient en arrière dans la source
- ◆ Revient à la position initiale
- ◆ Revient à la position initiale dans le dossier
- ◆ Reçoit des données d'une socket connectée
- ◆ Reçoit des données d'une socket, connectée ou pas
- ◆ Reçoit tous les objets snmp d'un agent
- ◆ Reçoit un message depuis une file de messages
- ◆ Reçoit un objet snmp
- ◆ Rotation de la transformation courante
- ◆ Rouvre une connexion mcal
- ◆ Réactive l'ancienne fonction de gestion d'exceptions
- ◆ Réactive l'ancienne fonction de gestion des erreurs
- ◆ Récupère certaines valeurs
- ◆ Récupère certains paramètres
- ◆ Récupère des informations d'en-tête d'une base de données dbase
- ◆ Récupère des informations sur le client maxdb
- ◆ Récupère des informations sur le dernière requête exécutée
- ◆ Récupère l'empreinte d'un serveur distant
- ◆ Récupère l'heure de la dernière modification d'un fichier
- ◆ Récupère l'heure de modification de l'inode d'un fichier
- ◆ Récupère l'inode d'un fichier
- ◆ Récupère l'itérateur interne
- ◆ Récupère la clé courante
- ◆ Récupère la date et l'heure du dernier accès à un fichier
- ◆ Récupère la largeur d'un glyphe
- ◆ Récupère la largeur d'une chaîne de caractères
- ◆ Récupère la ligne suivante d'un jeu de résultat pdo
- ◆ Récupère la miniature d'une image tiff ou jpeg
- ◆ Récupère la position du champs courant pour un pointeur de résultat
- ◆ Récupère la profondeur courante de la récursivité de l'itérateur
- ◆ Récupère la taille d'un fichier
- ◆ Récupère la valeur d'un paramètre
- ◆ Récupère la valeur de l'élément courant
- ◆ Récupère la version d'apache
- ◆ Récupère la version d'un tag id3
- ◆ Récupère le code erreur de l'analyseur xml
- ◆ Récupère le dernier numéro d'erreur d'ingres généré

- ◆ Récupère le décalage horaire par défaut utilisé par toutes les fonctions date/heure dans un script
- ◆ Récupère le groupe d'un fichier
- ◆ Récupère le nom court d'un champs id3v2
- ◆ Récupère le nom d'un glyphe
- ◆ Récupère le nom de la colonne dans la réponse délimitée par virgule
- ◆ Récupère le nom du curseur pour une ressource donnée
- ◆ Récupère le nom long d'un champs id3v2
- ◆ Récupère le nom pour un id de genre
- ◆ Récupère le nombre de champs dans un résultat
- ◆ Récupère le nombre de lignes affectées par la dernière opération maxdb
- ◆ Récupère le nombre de lignes dans un résultat
- ◆ Récupère le propriétaire d'un fichier
- ◆ Récupère le sqlstate associé lors de la dernière opération sur la requête
- ◆ Récupère le statut courant du système
- ◆ Récupère le tampon complet contenant les données ps générées
- ◆ Récupère le type d'un fichier
- ◆ Récupère les données brutes délimitées par virgule retournées par mcve
- ◆ Récupère les données métas d'un champ unique
- ◆ Récupère les entrées des éléments principaux depuis la base de données kerberos
- ◆ Récupère les géométries d'une chaîne de caractères
- ◆ Récupère les informations sur l'erreur associée lors dernière opération sur la requête
- ◆ Récupère les informations sur une relation
- ◆ Récupère les méta-données pour un seul champ
- ◆ Récupère les noms d'utilisateurs visibles
- ◆ Récupère les options d'un processeur xsl donné
- ◆ Récupère les permissions d'un fichier
- ◆ Récupère les propriétés du buffer openal
- ◆ Récupère les résultats depuis la requête préparée dans les variables liées
- ◆ Récupère les éléments par leur espace de noms et leur localname
- ◆ Récupère tous les en-têtes de la requête http
- ◆ Récupère tous les en-têtes de réponse http
- ◆ Récupère tous les en-têtes http de la requête
- ◆ Récupère tous les éléments principaux depuis la base de données kerberos
- ◆ Récupère toutes les directives depuis la base de données kerberos
- ◆ Récupère toutes les informations stockées dans un tag id3
- ◆ Récupère toutes les lignes d'une colonne de résultats particulière en tant que tableau
- ◆ Récupère toutes les valeurs possibles du genre
- ◆ Récupère un attribut d'une connexion à une base de données
- ◆ Récupère un attribut de requête
- ◆ Récupère un attribut étendu
- ◆ Récupère un identifiant unique
- ◆ Récupère un ienumvariant (déconseillé)
- ◆ Récupère un jeu de résultats depuis une requête préparée
- ◆ Récupère un message d'erreur significatif pour la dernière erreur générée
- ◆ Récupère un paramètre
- ◆ Récupère un paramètre spécial de réponse
- ◆ Récupère un tableau contenant les noms de variables courantes de portées locales
- ◆ Récupère un élément du serveur de cache
- ◆ Récupère une cellule spécifique d'une réponse délimitée par virgule avec le nom de colonne
- ◆ Récupère une cellule spécifique d'une réponse délimitée par virgule avec le numéro de colonne
- ◆ Récupère une ligne de résultat dans un tableau numérique, associatif ou les deux

- ◆ Récupère une ligne de résultat dans un objet
 - ◆ Récupère une ligne de résultat dans un tableau énuméré
 - ◆ Récupère une ligne de résultat dans un tableau associatif
 - ◆ Récupère une ligne de résultat en tant que tableau énuméré
 - ◆ Récupère une ligne de résultat ingres dans un tableau
 - ◆ Récupère une ligne de résultat sous forme de tableau associatif
 - ◆ Récupère une ligne de résultat sous forme de tableau indexé
 - ◆ Récupère une ligne du résultat d'une requête dans un tableau associatif
 - ◆ Récupère une liste d'attributs étendus
 - ◆ Récupère une propriété d'auditeur
 - ◆ Récupère une propriété de source openal
 - ◆ Récupère une valeur
 - ◆ Récupération d'un id pour un genre
 - ◆ Réduit itérativement un tableau
 - ◆ Réinitialise une requête préparée
 - ◆ Réouvre un flux icap pour un nouveau calendrier
 - ◆ Réouvre un flux imap vers une nouvelle boîte aux lettres
 - ◆ Répare un fichier et le renvoie en tant que chaîne
 - ◆ Répare une chaîne html en utilisant un fichier de configuration optionnel
 - ◆ Répète une chaîne
 - ◆ Réserve un sémaphore
 - ◆ Résolution dns d'une adresse ip
 - ◆ Résoud le chemin réel d'un chemin fourni
 - ◆ Résoud les informations d'hôte pour une relation
 - ◆ Rétablit l'ancien environnement graphique pdf
 - ◆ Rétablit le terminal sauvé
 - ◆ Rétablit les clés soft label
 - ◆ Rétablit un écran depuis un fichier de sauvegarde
- S
- ◆ S'identifie auprès du serveur hyperwave
 - ◆ S'identifie sur un serveur cyrus imap
 - ◆ Sauve dans un fichier
 - ◆ Sauve des données dans un lob oracle
 - ◆ Sauve l'environnement courant
 - ◆ Sauve l'environnement graphique courant
 - ◆ Sauve l'état du terminal
 - ◆ Sauve la configuration courante dans un fichier
 - ◆ Sauve le dictionnaire personnel dans un fichier
 - ◆ Sauve le mode de terminal (shell)
 - ◆ Sauve le mode du terminal
 - ◆ Sauve un écran dans un fichier
 - ◆ Sauve une clé au format ascii dans un fichier
 - ◆ Sauvegarde l'arbre interne xml dans un fichier
 - ◆ Sauvegarde l'arbre interne xml dans une chaîne de caractères
 - ◆ Sauvegarde l'espace de nom donné dans le contexte xpath passé
 - ◆ Sauvegarde l'espace de nom donné dans le contexte xpath passé
 - ◆ Sauvegarde la position
 - ◆ Sauvegarde le contexte courant
 - ◆ Sauvegarde le document interne dans une chaîne en utilisant un formattage html
 - ◆ Sauvegarde les informations de débogage dans un log
 - ◆ Sauvegarde un document fdf
 - ◆ Sauvegarde un document interne dans un fichier en utilisant un formattage html
 - ◆ Scanne les données du fichier et extrait tous les fichiers encodés qui s'y trouvent
 - ◆ Scelle des données

- ◆ Scinde un code source en éléments de base
- ◆ Scinde une chaîne
- ◆ Scinde une chaîne en tableau avec une expression rationnelle multi-octets
- ◆ Scinde une chaîne en un tableau, grâce à une expression rationnelle
- ◆ Scinde une chaîne en un tableau, grâce à une expression rationnelle
- ◆ Scrolle le contenu de la fenêtre vers le haut ou le bas, sans changer la position courante
- ◆ Se connecte à un serveur cyrus imap
- ◆ Se connecte à un serveur ldap
- ◆ Se déplace vers la prochaine entrée
- ◆ Se déplace vers la prochaine entrée
- ◆ Se déplace vers la prochaine entrée
- ◆ Se positionne sur le numéro de ligne précédent du jeu de résultats
- ◆ Se positionne sur une ligne arbitraire dans le jeu de résultats
- ◆ Signe du nombre gmp
- ◆ Signe les données
- ◆ Signe un csr avec un autre certificat
- ◆ Signe un message s/mime
- ◆ Similaire à define(), mais permet aussi la définition dans définitions de classe
- ◆ Sinus
- ◆ Sinyus hyperbolique
- ◆ Sonde le rpl
- ◆ Soumet une requête msql
- ◆ Souscrit à une boîte aux lettres
- ◆ Soustraction de 2 nombres gmp
- ◆ Soustrait la valeur du variant de droite de la valeur de celui de gauche et retourne le résultat
- ◆ Soustrait un nombre de grande taille à un autre
- ◆ Spécifie la base d'une session yaz
- ◆ Spécifie la méthode avec laquelle les valeurs snmp seront retournées
- ◆ Spécifie la nouvelle clé primaire d'une relation
- ◆ Spécifie la syntaxe de lecture des lignes
- ◆ Spécifie le jeu de caractères utilisé pour les messages du domaine domain
- ◆ Spécifie le nombre maximal de résultats à lire
- ◆ Spécifie le schéma de lecture
- ◆ Spécifie le type d'éléments yaz à lire
- ◆ Spécifie un autre descripteur de fichier pour la vérification à la volée des données saisies
- ◆ Spécifie une fonction utilisateur comme gestionnaire d'erreurs
- ◆ Spécifie une fonction à capturer et/ou traiter la sortie à partir d'un runkit sandbox
- ◆ Spécifie une valeur à un attribut
- ◆ Statistiques réseau irc
- ◆ Statue un fichier sur un système de fichiers distant
- ◆ Statue un lien symbolique
- ◆ Stock un enregistrement dans une base de données paradox
- ◆ Stocke des données dans le serveur de cache
- ◆ Stocke un message d'erreur
- ◆ Stoppe l'exécution du compilateur
- ◆ Stoppe l'interpréteur et attend un cr depuis le socket
- ◆ Stoppe le script pendant une durée spécifiée
- ◆ Stoppe le serveur web virtuel
- ◆ Stoppe le service iis identifié par serviceid
- ◆ Stoppe une base de données fbsql
- ◆ Suggère l'orthographe d'un mot [obsolète]

- ◆ Suggère une orthographe
- ◆ Supprime ...
- ◆ Supprime dynamiquement la méthode donnée
- ◆ Supprime la ligne courante et retourne la nouvelle ligne courante
- ◆ Supprime la récurrence de la structure globale
- ◆ Supprime la protection d'une chaîne de type bytea
- ◆ Supprime le serveur iis représenté par serverinstance
- ◆ Supprime les anti-slash d'une chaîne
- ◆ Supprime les balises html et php d'une chaîne
- ◆ Supprime les derniers éléments d'une collection oracle
- ◆ Supprime les espaces (ou d'autres caractères) de début de chaîne
- ◆ Supprime les espaces (ou d'autres caractères) de fin de chaîne
- ◆ Supprime les espaces (ou d'autres caractères) en début et fin de chaîne
- ◆ Supprime tous les événements marqués pour l'effacement
- ◆ Supprime toutes les lignes d'une relation
- ◆ Supprime toutes les ressources d'un document postscript
- ◆ Supprime un attribut
- ◆ Supprime un attribut hyperwave
- ◆ Supprime un calendrier
- ◆ Supprime un clé publique autorisée
- ◆ Supprime un fils de la liste des enfants
- ◆ Supprime un fils de la liste des noeuds fils
- ◆ Supprime un filtre d'un flux
- ◆ Supprime un flag (drapeau) sur un message
- ◆ Supprime un gestionnaire d'url
- ◆ Supprime un objet blob
- ◆ Supprime un objet char
- ◆ Supprime un objet d'une animation
- ◆ Supprime un objet d'une animation
- ◆ Supprime un objet dans un sprite
- ◆ Supprime un objet slob
- ◆ Supprime un segment de mémoire partagée sous unix
- ◆ Supprime un utilisateur de la liste des utilisateurs indésirables
- ◆ Supprime une base de données frontbase
- ◆ Supprime une base de données ibase
- ◆ Supprime une base lotus notes
- ◆ Supprime une couleur d'une image
- ◆ Supprime une sous-chaîne dans le noeud
- ◆ Supprime une transaction
- ◆ Supprime une transaction spécifiée de la structure mcve_conn
- ◆ Supprime une variable de la session
- ◆ Suspend le contexte spécifié
- ◆ Suspend temporairement le monitoring
- ◆ Symbole de jacobi
- ◆ Symbole de legendre
- ◆ Synchronise avec le serveur postgresql
- ◆ Synchronise une base de données dba
- ◆ Synonyme de checkdnsrr
- ◆ Synonyme de getmxrr
- ◆ Synonyme de odbc_exec
- ◆ Sélectionne la base de données ms sql
- ◆ Sélectionne la forme de départ
- ◆ Sélectionne la forme de fin
- ◆ Sélectionne la largeur et hauteur du champ

- ◆ Sélectionne la police courante
- ◆ Sélectionne la police courante et sa taille
- ◆ Sélectionne le mode non bloquant d'un descripteur de fichier
- ◆ Sélectionne un format d'affichage pour les messages irc
- ◆ Sélectionne un stylo
- ◆ Sélectionne une base de données frontbase
- ◆ Sélectionne une base de données msql
- ◆ Sélectionne une base de données mysql
- ◆ Sélectionne une base de données par défaut pour les requêtes
- ◆ Sélectionne une base de données pour les requêtes
- ◆ Sélectionne une base de données sybase
- ◆ Sélectionne une brosse
- ◆ Sélectionne une nouvelle zone pour un dessin ultérieur
- ◆ Sélectionne une police de caractères
- ◆ Sépare le nom du fichier et le nom du dossier
- ◆ Sépare les différents composants d'un dn
- ◆ Sépare un tableau en tableaux de taille inférieure
- T
- - ◆ Taille d'un document
 - ◆ Tangente
 - ◆ Tangente hyperbolique
 - ◆ Termine l'action courante
 - ◆ Termine l'enfin ccvs et libère les ressources
 - ◆ Termine la définition de symbole
 - ◆ Termine la définition du bouton flash courant
 - ◆ Termine la lecture de ressources oracle
 - ◆ Termine la page pdf courante
 - ◆ Termine la souscription à une boîte aux lettres
 - ◆ Termine le chemin et dessine les bords
 - ◆ Termine le chemin pdf courant
 - ◆ Termine le chemin, dessine les bords et remplit la forme
 - ◆ Termine le mode standout de la fenêtre
 - ◆ Termine le monitoring
 - ◆ Termine le pattern pdf
 - ◆ Termine le processus apache après cette requête
 - ◆ Termine le script courant
 - ◆ Termine le suivi d'une connexion postgresql
 - ◆ Termine le template pdf
 - ◆ Termine un chiffrement
 - ◆ Termine un document
 - ◆ Termine un flux imap
 - ◆ Termine un jeu de sessions curl
 - ◆ Termine un modèle
 - ◆ Termine un motif
 - ◆ Termine un processus ouvert par proc_open
 - ◆ Termine une connexion postgresql
 - ◆ Termine une page
 - ◆ Termine une page
 - ◆ Termine une section de texte
 - ◆ Termine une séquence maxdb_stmt_send_long_data
 - ◆ Test rpl
 - ◆ Test si l'implémentation dom implémente une fonctionnalité spécifique
 - ◆ Test une connexion à un serveur ou tente de se reconnecter si la connexion a été interrompue

- ◆ Teste la fin du fichier
- ◆ Teste la fin du lob oracle
- ◆ Teste le chiffrement par blocs d'un algorithme
- ◆ Teste le chiffrement par blocs d'un mode
- ◆ Teste si la valeur d'une colonne oracle est null
- ◆ Teste si le mode retourne les données par blocs
- ◆ Teste si un champ est annulable
- ◆ Teste si un champ postgresql est à null
- ◆ Teste un mode
- ◆ Teste un module ouvert
- ◆ Tourne la forme
- ◆ Tourne un objet en angle absolu
- ◆ Traite le contexte spécifié
- ◆ Traite un fichier de configuration
- ◆ Traite un fichier php important fonctions et définitions de classes. écrasement où applicable
- ◆ Traite un message com en attendant jusqu'à timeoutms millisecondes
- ◆ Transfert un jeu de résultats depuis la dernière requête
- ◆ Transfert un jeu de résultats depuis une requête préparée
- ◆ Transforme des données xml avec xslt
- ◆ Transforme en un document dom
- ◆ Transforme en uri
- ◆ Transforme en xml
- ◆ Transforme les coordonnées
- ◆ Transforme les coordonnées d'une fenêtre
- ◆ Transforme un objet domelement en un objet simplexmlement
- ◆ Transforme un texte anglais en timestamp
- ◆ Transforme une chaîne ascii en ebcdic
- ◆ Transforme une chaîne ebcdic en ascii
- ◆ Transforme une liste de variables en tableau
- ◆ Transforme une représentation de clé dba par chaîne en une représentation par tableau
- ◆ Transfère des fonds de chez le marchand vers le possesseur de la carte de crédit
- ◆ Transfère des fonds de chez le possesseur de la carte de crédit vers le marchand
- ◆ Transfère un jeu de résultats à partir de la dernière requête
- ◆ Translate la transformation courante
- ◆ Traverse la carte et applique une fonction sur chaque entrée
- ◆ Tri multi-dimensionnel de tableaux
- ◆ Trie des messages
- ◆ Trie les entrées d'un résultat ldap
- ◆ Trie un résultat avec une fonction utilisateur
- ◆ Trie un tableau
- ◆ Trie un tableau avec l'algorithme à "ordre naturel"
- ◆ Trie un tableau avec l'algorithme à "ordre naturel" insensible à la casse
- ◆ Trie un tableau en ordre inverse
- ◆ Trie un tableau en ordre inverse
- ◆ Trie un tableau en sens inverse et suivant les clés
- ◆ Trie un tableau en utilisant une fonction de callback
- ◆ Trie un tableau en utilisant une fonction de comparaison
- ◆ Trie un tableau et conserve l'association des index
- ◆ Trie un tableau par ses clés en utilisant une fonction de callback
- ◆ Trie un tableau suivant les clés
- ◆ Tronque un fichier
- ◆ Tronque un fichier

- ◆ Tronque un lob oracle
- ◆ Tronque une chaîne
- ◆ Trouve la dernière occurrence d'un caractère dans une chaîne
- ◆ Trouve la position d'un caractère dans une chaîne
- ◆ Trouve la position de la dernière occurrence d'un élément dans une chaîne, à partir d'un offset
- ◆ Trouve la position de la dernière occurrence d'un caractère dans une chaîne
- ◆ Trouve la position de la dernière occurrence d'une chaîne dans une autre de façon insensible à la casse
- ◆ Trouve la position de la première occurrence de needle dans haystack
- ◆ Trouve la première occurrence dans une chaîne
- ◆ Trouve le nombre de transactions d'un certain type stockées dans le système
- ◆ Trouve le premier segment de chaîne
- ◆ Trouve les enfants d'une node
- ◆ Trouve toutes les sessions avec un nom et une valeur
- ◆ Trouve un segment de chaîne ne contenant pas certains caractères
- ◆ Type de données d'un champ
- ◆ Télécharge un fichier depuis un serveur ftp
- ◆ Télécharge un fichier via ftp dans un fichier local
- **U**
 - ◆ Un autre moyen pour définir une couche mask
 - ◆ Utilisation des ressources
 - ◆ Utilisation ou non du gestionnaire d'erreurs soap et retourne l'ancienne valeur
 - ◆ Utilise la version de swf (??)
 - ◆ Utilise un analyseur xml à l'intérieur d'un objet
 - ◆ Utilise une constante de réserve (?)
 - ◆ Utilise une variable php pour la phase de définition, dans une commande select oracle
 - ◆ Utiliser ou non les fonctions d'antialias
 - ◆ Utilisé pour établir une connexion sécurisée utilisant ssl
 - ◆ Utilisée pour établir une connexion sécurisée avec ssl
- **V**
 - ◆ Va à l'entrée simplexml suivante
 - ◆ Va à la prochaine image du sprite
 - ◆ Valeur absolue
 - ◆ Valeur absolue gmp
 - ◆ Valide la transaction courante
 - ◆ Valide la transaction courante
 - ◆ Valide la transaction sesam en cours
 - ◆ Valide le mot de passe d'un utilisateur pour un domaine
 - ◆ Valide les transactions oracle en cours
 - ◆ Valide ou non l'identifiant passé sur n'importe quelle transaction qu'il est passé
 - ◆ Valide toutes les modifications de la base de données kerberos, et quitte la connexion ouverte du serveur d'administration kerberos
 - ◆ Valide un document en se basant sur sa dtd
 - ◆ Valide un document selon un schéma
 - ◆ Valide un document selon un schéma
 - ◆ Valide une date
 - ◆ Valide une date grégorienne
 - ◆ Valide une heure
 - ◆ Valide une transaction
 - ◆ Valide une transaction
 - ◆ Valide une transaction
 - ◆ Valide une transaction ibase

- ◆ Valide une transaction ibase sans la refermer
- ◆ Valide une transaction odbc
- ◆ Valide une transaction oracle
- ◆ Valide une transaction ovrimos
- ◆ Valide une version autre que la dernière
- ◆ Verrouille le fichier
- ◆ Verrouille un objet
- ◆ Verrouille une session
- ◆ Version insensible à la casse de str_replace
- ◆ Version insensible à la casse de strpos
- ◆ Version plurielle de dcgettext
- ◆ Version plurielle de dgettext
- ◆ Version plurielle de gettext
- ◆ Vide le buffer d'entrée du clavier
- ◆ Vide le buffer d'erreur libxml
- ◆ Vide le tampon, le retourne en tant que chaîne et stoppe la mise en tampon
- ◆ Vide les buffers après détection d'un caractère de signal
- ◆ Vide les buffers de sortie
- ◆ Vérifie si un objet ou une classe a une propriété
- ◆ Vérification tcpwrap
- ◆ Vérifie l'existence d'une constante
- ◆ Vérifie l'usage d'un certificat
- ◆ Vérifie l'état de la connexion
- ◆ Vérifie la connexion mnogosearch avec le document stocké
- ◆ Vérifie la présence des touches de fonctions sur le clavier
- ◆ Vérifie la signature d'un message s/mime
- ◆ Vérifie la syntaxe php (et exécute) d'un fichier spécifique
- ◆ Vérifie la syntaxe php d'un fichier spécifié
- ◆ Vérifie la syntaxe php de code php spécifié
- ◆ Vérifie le courrier de la boîte aux lettres courante
- ◆ Vérifie le status d'une facture
- ◆ Vérifie les capacités d'insertion et d'effacement
- ◆ Vérifie les capacités d'insertion et d'effacement
- ◆ Vérifie ou non le certificat du serveur ssl
- ◆ Vérifie qu'un caractère est un caractère de contrôle
- ◆ Vérifie qu'un caractère représente un nombre hexadécimal
- ◆ Vérifie qu'un identifiant d'objet est dans un groupe
- ◆ Vérifie qu'une chaîne contient de la ponctuation
- ◆ Vérifie qu'une chaîne est alpha-numérique
- ◆ Vérifie qu'une chaîne est alphabétique
- ◆ Vérifie qu'une chaîne est en majuscules
- ◆ Vérifie qu'une chaîne est en minuscules
- ◆ Vérifie qu'une chaîne est imprimable
- ◆ Vérifie qu'une chaîne est imprimable
- ◆ Vérifie qu'une chaîne est un entier
- ◆ Vérifie qu'une chaîne n'est faite que de caractères blancs
- ◆ Vérifie qu'une classe a été définie
- ◆ Vérifie qu'une clé dba existe
- ◆ Vérifie que l'année est bissextile
- ◆ Vérifie que la méthode existe pour une classe
- ◆ Vérifie que le flux imap est toujours actif
- ◆ Vérifie que le terminal supporte les couleurs
- ◆ Vérifie s'il y a d'autres jeux de résultats mysql disponibles
- ◆ Vérifie s'il y a encore des résultats de disponibles depuis de multiples requêtes

- ◆ Vérifie si des événements fam sont disponibles
- ◆ Vérifie si l'analyseur rpl est activé
- ◆ Vérifie si l'api soap a échoué
- ◆ Vérifie si l'attribut est un identifiant défini
- ◆ Vérifie si l'entrée courante est un dossier et n'est pas '.' ou '..'
- ◆ Vérifie si l'itérateur interne a un élément suivant valide
- ◆ Vérifie si l'élément courant de l'itérateur interne a un fils
- ◆ Vérifie si l'élément courant de l'itérateur interne a un fils
- ◆ Vérifie si l'élément courant est valide
- ◆ Vérifie si l'élément courant est valide
- ◆ Vérifie si l'élément courant est valide
- ◆ Vérifie si la connexion postgresql est occupée
- ◆ Vérifie si la fonctionnalité est disponible pour la version spécifiée
- ◆ Vérifie si la position courante est valide
- ◆ Vérifie si la réponse est délimitée par virgule
- ◆ Vérifie si la transaction a été complétée
- ◆ Vérifie si la transaction fut complétée correctement
- ◆ Vérifie si le noeud possède des enfants
- ◆ Vérifie si le noeud possède un attribut
- ◆ Vérifie si le parseur rpl est activé
- ◆ Vérifie si le répertoire contient encore des entrées
- ◆ Vérifie si le tampon de sortie est vide
- ◆ Vérifie si le terminal peut changer de couleurs
- ◆ Vérifie si plusieurs résultats sont disponibles
- ◆ Vérifie si un attribut est spécifié
- ◆ Vérifie si un attribut est éditable
- ◆ Vérifie si un attribut existe
- ◆ Vérifie si un attribut existe
- ◆ Vérifie si un attribut existe
- ◆ Vérifie si un fichier existe
- ◆ Vérifie si un jeu de caractères est supporté par mnogosearch
- ◆ Vérifie si un noeud a des attributs
- ◆ Vérifie si un noeud a des fils
- ◆ Vérifie si un noeud est blanc
- ◆ Vérifie si un système de fichier supporte les attributs étendus
- ◆ Vérifie si un tableau contient d'autres entrées
- ◆ Vérifie si une assertion est fausse
- ◆ Vérifie si une clé existe dans un tableau
- ◆ Vérifie si une clé privée correspond à un certificat
- ◆ Vérifie si une interface a été définie
- ◆ Vérifie si une ressource simplexml contient d'autres entrées
- ◆ Vérifie si une variable est enregistrée dans la session
- ◆ Vérifie un mot
- ◆ Vérifie un mot [obsolète]
- ◆ Vérifie un mot sans en changer la casse et sans essayer de supprimer les espaces aux extrémités [obsolète]
- ◆ Vérifie une signature
- É
- - ◆ Écrit des données dans le fichier avec la possibilité de tronquer sa longueur
 - ◆ Écrit du texte
 - ◆ Écrit du texte à la position donnée
 - ◆ Écrit le résultat d'un transformation xslt dans un fichier
 - ◆ Écrit le résultat d'un transformation xslt dans une chaîne
 - ◆ Écrit le texte donné dans l'objet swftext à la position du crayon, en utilisant la police.

la hauteur, l'espacement et la couleur courante

- ◆ Écrit un cadre
- ◆ Écrit un cadre
- ◆ Écrit un glyphe
- ◆ Écriture de texte dans une boîte
- ◆ Établit une connexion postgresql
- ◆ Établit une connexion postgresql persistante
- ◆ Établit une connexion à mcve
- ◆ Éteint la bibliothèque payflow pro

13.3 Index des exemples

- ◆ A
- ◆ \$ ◇
- ◆ ◇ \$this peut être indéfinie : cas d'une méthode statique
- ◆ -
- ◆ ◇ Exemple avec __autoload
- ◆ ◇ Exemple avec __halt_compiler
- ◆ A
- ◆ ◇ Exemple avec abs
- ◆ ◇ Accès aux données de formulaire fdf
- ◆ ◇ Accès aux valeurs rgb
- ◆ ◇ Accès aux variables parentes
- ◆ ◇ Accéder aux données du formulaire
- ◆ ◇ Accéder à un tableau dans des doubles guillemets
- ◆ ◇ Activer l'extension bzip2 pour php-windows
- ◆ ◇ Exemple avec addaction.php">swfbutton->addaction
- ◆ ◇ exemple avec addslashes et charlist (2)
- ◆ ◇ Exemple avec addfill.php">swfshape->addfill
- ◆ ◇ Exemple avec addslashes
- ◆ ◇ Affectation d'une colonne de type "champs multiples"
- ◆ ◇ Affichage de texte
- ◆ ◇ Affichage de texte avec cpdf_begin_text
- ◆ ◇ Affichage de toutes les options avec leur documentation et leur valeur par défaut
- ◆ ◇ Affichage de toutes les permissions
- ◆ ◇ Affichage des noeuds
- ◆ ◇ Affichage sous la forme d'une table html
- ◆ ◇ Affiche la liste des attributs d'une entrée
- ◆ ◇ Affiche les champs de errorinfo() pour une connexion pdo_odbc sur une base de données db2
- ◆ ◇ Affiche les champs de errorinfo() pour une connexion pdo_odbc sur une base de données db2
- ◆ ◇ Affiche les informations d'en-têtes pour un fichier de base de données dbase
- ◆ ◇ Affiche les noms de tous les attributs étendus d'un fichier
- ◆ ◇ Afficher des données issues d'un formulaire
- ◆ ◇ Afficher sans l'aide de echo
- ◆ ◇ Afficher tous les objet avec snmpwalk
- ◆ ◇ Afficher une erreur sesam
- ◆ ◇ Afficher une somme de contrôle crc32
- ◆ ◇ Afficher une structure xml
- ◆ ◇ Exemple avec aggregate_info
- ◆ ◇ Ajout d'entrées dans un tableau
- ◆ ◇ Ajout d'un attribut

- ◇ [Ajout d'un fils](#)
- ◇ [Ajout d'une clé publique avec ssh2_publickey_add](#)
- ◇ [Ajout d'une note à un document pdf avec la pdflib](#)
- ◇ [Ajout dans le chemin d'inclusion](#)
- ◇ [Ajouter du code javascript à un document fdf](#)
- ◇ [Ajouter un noeud fils avec dom xml \(exemple 3\)](#)
- ◇ [Ajouter un noeud fils dom xml](#)
- ◇ [Ajouter une mise en relief](#)
- ◇ [Alimentation de sortie vers une variable](#)
- ◇ [Alternatives à scandir pour php 4](#)
- ◇ [Analyse ccl](#)
- ◇ [Analyser le résultat de socket_select](#)
- ◇ [Annulation d'une transaction](#)
- ◇ [Annulation d'une transaction sesam](#)
- ◇ [Annule une transaction](#)
- ◇ [Exemple avec apache_get_modules](#)
- ◇ [Exemple avec apache_get_version](#)
- ◇ [Exemple avec apache_getenv](#)
- ◇ [Exemple avec apache_lookup_uri](#)
- ◇ [Exemple avec apache_request_headers](#)
- ◇ [Exemple avec apache_response_headers](#)
- ◇ [Exemple avec apd_callstack](#)
- ◇ [Exemple avec apd_clunk](#)
- ◇ [Exemple avec apd_continue](#)
- ◇ [Exemple avec apd_croak](#)
- ◇ [Exemple avec apd_dump_function_table](#)
- ◇ [Exemple avec apd_dump_persistent_resources](#)
- ◇ [Exemple avec apd_dump_regular_resources](#)
- ◇ [Exemple avec apd_echo](#)
- ◇ [Exemple avec apd_get_active_symbols](#)
- ◇ [Exemple avec apd_set_pprof_trace](#)
- ◇ [Exemple avec apd_set_session](#)
- ◇ [Exemple avec apd_set_session_trace](#)
- ◇ [Exemple avec apd_set_socket_session_trace](#)
- ◇ [Appel d'une méthode parent](#)
- ◇ [Appel d'une procédure stockée avec un paramètre d'entrée/sortie](#)
- ◇ [Appel une procédure stockée avec un paramètre inout](#)
- ◇ [Argument de fonction de lecture](#)
- ◇ [Exemple avec array_change_key_case](#)
- ◇ [Exemple avec array_chunk](#)
- ◇ [Exemple avec array_combine](#)
- ◇ [Exemple avec array_count_values](#)
- ◇ [Exemple avec array_diff](#)
- ◇ [Exemple avec array_diff_assoc](#)
- ◇ [Exemple avec array_diff_key](#)
- ◇ [Exemple avec array_diff_uassoc](#)
- ◇ [Exemple avec array_diff_ukey](#)
- ◇ [Exemple avec array_fill](#)
- ◇ [array_filter sans callback](#)
- ◇ [Exemple avec array_flip : collision](#)
- ◇ [Exemple avec array_intersect](#)
- ◇ [Exemple avec array_intersect_assoc](#)
- ◇ [Exemple avec array_intersect_key](#)
- ◇ [Exemple avec array_intersect_uassoc](#)

- ◇ [Exemple avec array_intersect_ukey](#)
- ◇ [array_key_exists et isset](#)
- ◇ [Exemple avec array_keys](#)
- ◇ [Exemple avec array_merge en php 5](#)
- ◇ [Exemple avec array_merge_recursive](#)
- ◇ [Exemple avec array_pad](#)
- ◇ [Exemple avec array_pop](#)
- ◇ [Exemple avec array_product](#)
- ◇ [Exemple avec array_push](#)
- ◇ [Exemple avec array_rand](#)
- ◇ [Exemple avec array_reduce](#)
- ◇ [Exemple avec array_reverse](#)
- ◇ [Exemple avec array_search](#)
- ◇ [Exemple avec array_shift](#)
- ◇ [Exemple avec array_sum](#)
- ◇ [Exemple avec array_udiff](#)
- ◇ [Exemple avec array_udiff_assoc](#)
- ◇ [Exemple avec array_udiff_uassoc](#)
- ◇ [Exemple avec array_uintersect](#)
- ◇ [Exemple avec array_uintersect_assoc](#)
- ◇ [Exemple avec array_uintersect_uassoc](#)
- ◇ [array_unique et les types de valeurs](#)
- ◇ [Exemple avec array_walk](#)
- ◇ [Exemple avec array_walk_recursive](#)
- ◇ [Exemple avec arrayiterator::current](#)
- ◇ [Exemple avec arrayiterator::key](#)
- ◇ [Exemple avec arrayiterator::next](#)
- ◇ [Exemple avec arrayiterator::rewind](#)
- ◇ [Exemple avec arrayiterator::valid](#)
- ◇ [Exemple avec arrayobject::__construct](#)
- ◇ [Exemple avec arrayobject::getiterator](#)
- ◇ [Arrêter et démarrer apache une fois que php 4 est installé](#)
- ◇ [Exemple avec arsort](#)
- ◇ [Exemple avec asort](#)
- ◇ [Exemple avec aspell_check](#)
- ◇ [Exemple avec aspell_check_raw](#)
- ◇ [Exemple avec aspell_new](#)
- ◇ [Exemple avec aspell_suggest](#)
- ◇ [Assignment d'un objet](#)
- ◇ [Assignement d'une valeur par défaut](#)
- ◇ [Association d'objets](#)
- ◇ [Associativité](#)
- ◇ [Attaque d'un serveur de base de données \(mssql server\) - 2](#)
- ◇ [Attention aux erreurs de fractions](#)
- ◆ **B**
- ◆ [Balises d'ouvertures et de fermetures php](#)
- ◇ [Exemple avec base64_decode](#)
- ◇ [Exemple avec base64_encode](#)
- ◇ [Exemple avec base_convert](#)
- ◇ [Exemple avec basename](#)
- ◇ [Exemple avec bcadd](#)
- ◇ [Exemple avec bccomp](#)
- ◇ [Exemple avec bcddiv](#)
- ◇ [Exemple avec bcmath](#)

- ◇ [Exemple avec bcmath](#)
 - ◇ [Exemple avec bcompiler_load](#)
 - ◇ [Exemple avec bcompiler_load](#)
 - ◇ [Exemple avec bcompiler_parse_class](#)
 - ◇ [Exemple avec bcompiler_read](#)
 - ◇ [Exemple avec bcompiler_write_class](#)
 - ◇ [Exemple avec bcompiler_write_constant](#)
 - ◇ [Exemple avec bcompiler_write_file](#)
 - ◇ [Exemple avec bcompiler_write_footer](#)
 - ◇ [Exemple avec bcompiler_write_footer](#)
 - ◇ [Exemple avec bcompiler_write_function](#)
 - ◇ [Exemple avec bcompiler_write_functions_from_file](#)
 - ◇ [Exemple avec bcompiler_write_header](#)
 - ◇ [Exemple avec bcpow](#)
 - ◇ [Exemple avec bcpowmod](#)
 - ◇ [Exemple avec bcscale](#)
 - ◇ [Exemple avec bcsqrt](#)
 - ◇ [Exemple avec bcsub](#)
 - ◇ [Exemple avec bindec](#)
 - ◇ [Exemple avec bindtextdomain](#)
 - ◇ [Exemple avec bzerror](#)
 - ◇ [bzip2.compress et bzip2.decompress](#)
 - ◇ [Exemple avec bzopen](#)
 - ◇ [Exemple avec bzread](#)
 - ◇ [Exemple avec bzwrite](#)
- ◆ C
- ◆ [Exemple avec cal_days_in_month](#)
 - ◆ [Exemple avec cal_from_jd](#)
 - ◆ [cal_info example](#)
 - ◆ [Calcul du hachage et enregistrement d'un utilisateur de liste de diffusion](#)
 - ◆ [Calcule le md5 et le hmac, puis l'affiche comme un hexadécimal](#)
 - ◆ [Exemple avec call_user_func et une méthode de classe](#)
 - ◆ [Exemple avec call_user_func_array](#)
 - ◆ [Exemple avec call_user_method](#)
 - ◆ [Cas des colonnes de noms identiques avec sybase](#)
 - ◆ [Cas où toString est appelée](#)
 - ◆ [Exemple avec ceil](#)
 - ◆ [Ces lignes ajoutent le support mysql, curl et gd au sein de php](#)
 - ◆ [Cette ligne active le support de php au sein d'apache](#)
 - ◆ [Chapter.xml](#)
 - ◆ [Chargement d'un fichier via scp](#)
 - ◆ [Exemple avec chdir](#)
 - ◆ [Exemple avec checkdate](#)
 - ◆ [Exemple avec chr](#)
 - ◆ [Exemple avec chunk_split](#)
 - ◆ [Exemple avec class_implements](#)
 - ◆ [Exemple avec class_parents](#)
 - ◆ [Exemple avec classkit_import](#)
 - ◆ [Exemple avec classkit_method_add](#)
 - ◆ [Exemple avec classkit_method_copy](#)
 - ◆ [Exemple avec classkit_method_redefine](#)
 - ◆ [Exemple avec classkit_method_remove](#)
 - ◆ [Exemple avec classkit_method_rename](#)
 - ◆ [Exemple avec closedir](#)

- ◇ [Collection](#)
- ◇ [Collection](#)
- ◇ [Collection.xsl](#)
- ◇ [Comment gérer les valeurs retournées par dbx_fetch_row](#)
- ◇ [Comment retourner un flux depuis une fonction](#)
- ◇ [Comment utiliser le paramètre mode de chmod](#)
- ◇ [Comment utiliser les requêtes unbuffered](#)
- ◇ [Exemple avec compact](#)
- ◇ [Compactage d'une chaîne](#)
- ◇ [Comparaison d'objets composés en php 4](#)
- ◇ [Comparaison d'une empreinte à une valeur connue](#)
- ◇ [Comparer des tableaux](#)
- ◇ [Compilation cgi avec php \(unix\)](#)
- ◇ [Compression de données](#)
- ◇ [Comptage des colonnes](#)
- ◇ [Comptage des lignes retournées par une requête select](#)
- ◇ [Compter le nombre de passages d'un utilisateur sur une page](#)
- ◇ [Configuration ccl](#)
- ◇ [Configuration de php.ini pour les utilisateurs de sjis](#)
- ◇ [configuration des gestionnaires de schéma pour xslt](#)
- ◇ [Configuration explicite](#)
- ◇ [Configuration interbase](#)
- ◇ [Configuration isapi de sambar](#)
- ◇ [Configuration pour utiliser simultanément php 3 et 4](#)
- ◇ [Configuration sesam](#)
- ◇ [Connexion à un serveur informix](#)
- ◇ [Connexion anonyme à un serveur ldap](#)
- ◇ [connexion au serveur ovrimos sql server et sélection d'une table système](#)
- ◇ [connexion à un serveur ovrimos sql server et préparation d'une requête](#)
- ◇ [Connexion à une base sesam](#)
- ◇ [Connexions persistantes](#)
- ◇ [Exemple avec constant](#)
- ◇ [Constructeur de classe](#)
- ◇ [Contrôler l'application des filtres](#)
- ◇ [Conversion automatique indéfinie\(2\)](#)
- ◇ [Conversion d'un nombre gmp en chaîne](#)
- ◇ [Conversion des fichiers .htaccess pour php 4](#)
- ◇ [Conversion en booléen](#)
- ◇ [Exemple avec convert_uudecode](#)
- ◇ [Exemple avec convert_uuencode](#)
- ◇ [Convertir une chaîne xml](#)
- ◇ [Copie d'une ressource curl](#)
- ◇ [Copie dans un tableau par référence](#)
- ◇ [Exemple avec copy](#)
- ◇ [Exemple avec cos](#)
- ◇ [Exemple avec count_chars](#)
- ◇ [Critères de tri](#)
- ◇ [Cryptage d'une valeur avec tripledes sous 2.4.x en mode ecb](#)
- ◇ [Création d'un xsltprocessor](#)
- ◇ [Création d'un camembert en 3d](#)
- ◇ [Création d'un document](#)

- ◇ [Création d'un document avec une dtd attaché](#)
- ◇ [Création d'un document pdf en mémoire](#)
- ◇ [Création d'un dossier sur un serveur distant](#)
- ◇ [Création d'un en-tête de document html](#)
- ◇ [Création d'un en-tête de document html avec `dump-file.php`>domdocument->dump_file](#)
- ◇ [Création d'un en-tête de document html avec `dump-mem.php`>domdocument->dump_mem](#)
- ◇ [Création d'un en-tête de document html avec `html-dump-mem.php`>domdocument->html_dump_mem](#)
- ◇ [Création d'un fichier de base de données dbase](#)
- ◇ [création d'un fichier flash simple, basé sur une entrée de l'utilisateur, et sauvegarde dans une base.](#)
- ◇ [Création d'un fichier gzip](#)
- ◇ [création d'un flux d'image gd. et affichage](#)
- ◇ [Création d'un lien symbolique](#)
- ◇ [Création d'un nombre gmp](#)
- ◇ [Création d'un nouveau domattr](#)
- ◇ [Création d'un nouveau domcomment](#)
- ◇ [Création d'un nouveau domdocument](#)
- ◇ [Création d'un nouveau domelement](#)
- ◇ [Création d'un nouveau dumentityreference](#)
- ◇ [Création d'un nouveau domprocessinginstruction](#)
- ◇ [Création d'un nouveau domtext](#)
- ◇ [Création d'un nouvel élément et insertion en tant que racine](#)
- ◇ [Création d'un nouvel élément et insertion en tant que racine](#)
- ◇ [Création d'un objet](#)
- ◇ [Création d'un tableau de tableaux](#)
- ◇ [Création d'une base de données paradox avec deux champs](#)
- ◇ [Création d'une image png avec php](#)
- ◇ [création d'une image gd et affichage de cette image](#)
- ◇ [Création d'une instance pdo en utilisant un alias](#)
- ◇ [Créer un attribut avec dom xml](#)
- ◇ [Créer un nouveau bloc de mémoire partagée shmop](#)
- ◇ [Exemple avec ctype_alnum \(en utilisant les locales par défaut\)](#)
- ◇ [Exemple avec ctype_alpha \(en utilisant les locales courantes\)](#)
- ◇ [Exemple avec ctype_cntrl](#)
- ◇ [Exemple avec ctype_digit](#)
- ◇ [Exemple avec ctype_graph](#)
- ◇ [Exemple avec ctype_lower \(en utilisant les locales courantes\)](#)
- ◇ [Exemple avec ctype_print](#)
- ◇ [Exemple avec ctype_punct](#)
- ◇ [Exemple avec ctype_space](#)
- ◇ [Exemple avec ctype_upper \(en utilisant la locale courante\)](#)
- ◇ [Exemple avec ctype_xdigit](#)
- ◇ [Exemple avec current](#)
- ◆ **D**
- ◆ [dans certains cas, les classes doivent être déclarées avant d'être utilisées](#)
- ◇ [Dans une url](#)
- ◇ [Exemple avec date_sunrise](#)
- ◇ [Exemple avec date_sunset](#)
- ◇ [Exemple avec dba_handlers](#)
- ◇ [Exemple avec dbase_numfields](#)
- ◇ [Exemple avec dbx_close](#)

- ◇ [Exemple avec dbx_compare](#)
 - ◇ [Exemple avec dbx_connect](#)
 - ◇ [Exemple avec dbx_error](#)
 - ◇ [Exemple avec dbx_escape_string](#)
 - ◇ [Exemple avec dbx_sort](#)
 - ◇ [Exemple avec debug_backtrace](#)
 - ◇ [Exemple avec debug_print_backtrace](#)
 - ◇ [Exemple avec decbin](#)
 - ◇ [Exemple avec dechex](#)
 - ◇ [Exemple avec decoct](#)
 - ◇ [Exemple avec deg2rad](#)
 - ◇ [Demarrer les règles de réécriture d'apache pour hyperwave](#)
 - ◇ [Dernier jour du mois suivant](#)
 - ◇ [Dessin d'un cercle avec imagearc](#)
 - ◇ [Dessiner un rectangle](#)
 - ◇ [Destruction d'une session avec session_destroy](#)
 - ◇ [Différents comportements de dirname](#)
 - ◇ [Exemple avec dir](#)
 - ◇ [Exemple avec disk_free_space](#)
 - ◇ [Exemple avec disk_total_space](#)
 - ◇ [Division de nombres gmp](#)
 - ◇ [Divisions entières](#)
 - ◇ [Exemple avec dl](#)
 - ◇ [Exemple avec dns_get_record et dns_any](#)
 - ◇ [Exemple avec domattr->isid\(\)](#)
 - ◇ [Exemple avec domdocument->getelementbyid\(\)](#)
 - ◇ [Exemple avec domdocument->xinclude\(\)](#)
 - ◇ [Exemple avec domxml_version](#)
 - ◇ [Exemple avec domxml_xslt_version](#)
 - ◇ [Déclaration d'une méthode](#)
 - ◇ [Déclaration de variables statiques](#)
 - ◇ [Déclaration des membres](#)
 - ◇ [Décoder des entités html](#)
 - ◇ [Décompression d'une chaîne](#)
 - ◇ [Définir une constante](#)
 - ◇ [Définition des attributs étendus dans un fichier .wav](#)
 - ◇ [Définition du mode de récupération](#)
 - ◇ [Définition d'une constante](#)
 - ◇ [Définition du débit des bauds sur un port série](#)
 - ◇ [Définition et utilisation d'une constante de classe](#)
 - ◇ [Définition simple d'une classe](#)
 - ◇ [Désactivation des guillemets magiques à l'exécution](#)
 - ◇ [Désactiver l'analyse des fichiers php dans un dossier avec .htaccess](#)
 - ◇ [Désagrégation d'objets](#)
 - ◇ [Détecter des variables non protégées avec e_all](#)
 - ◇ [Détecter tous les noms d'un formulaire fdf](#)
 - ◇ [détection automatique des formats d'image supportés avec imagetypes](#)
 - ◇ [Détection simple de fausses variables](#)
 - ◇ [Détermine la catégorie de l'erreur qui survient](#)
 - ◇ [Détermine quelles méthodes ont été négociées](#)
 - ◇ [Détruire une référence](#)
- ◆ E
- ◆ [Exemple avec easter_date](#)
 - ◆ [Exemple avec easter_days](#)

- ◇ [Echappement d'une chaîne complexe](#)
- ◇ [Ecrire des données dans un fichier compressé](#)
- ◇ [Ecrire un bloc de mémoire partagée](#)
- ◇ [Efface tous les attributs étendus d'un fichier](#)
- ◇ [Effacement d'un bloc de mémoire partagée](#)
- ◇ [Effacement d'un dossier sur un serveur distant](#)
- ◇ [Effacement d'un fichier](#)
- ◇ [Effacement d'un fils](#)
- ◇ [Effacer une ressource existante](#)
- ◇ [Éléments retournés par imap_headerinfo](#)
- ◇ [Emulation des register globals](#)
- ◇ [Encoder un fichier audio en ogg/vorbis](#)
- ◇ [Exemple avec end](#)
- ◇ [Enregistrement d'un fichier téléchargé](#)
- ◇ [Enregistrement d'une classe de filtre générique pour correspondre avec de multiples noms de filtres.](#)
- ◇ [Enumerer tous les messages d'erreur ldap](#)
- ◇ [Envoi d'une commande site à un serveur ftp](#)
- ◇ [Envoi de mail html](#)
- ◇ [Envoie d'un tableau de fichiers](#)
- ◇ [Equivalent de variables variables](#)
- ◇ [Exemple avec ereg](#)
- ◇ [Exemple avec eregi](#)
- ◇ [error_log examples](#)
- ◇ [Exemple avec error_reporting](#)
- ◇ [Exemple avec escapeshellarg](#)
- ◇ [Exemple avec escapeshellcmd](#)
- ◇ [Etendre la classe exception](#)
- ◇ [exemple avec eval - concaténation de texte](#)
- ◇ [Evaluer un document fdf](#)
- ◇ [Exemple avec exec](#)
- ◇ [Exemple avec mb_convert_case](#)
- ◇ [Exemple fbsql_num_rows](#)
- ◇ [Exemple ibase_connect](#)
- ◇ [Exemple imap_mime_header_decode](#)
- ◇ [Exemple mysql_db_name](#)
- ◇ [Exemple pg_send_prepare](#)
- ◇ [Exemple pspell_add_to_personal](#)
- ◇ [Exemple restore_include_path](#)
- ◇ [Exemple setline.php">swfshape->setline](#)
- ◇ [Exemple setname.php">swfdisplayitem->setname](#)
- ◇ [Exemple str_shuffle](#)
- ◇ [Exemple alternatif avec mysql_db_query](#)
- ◇ [Exemple alternatif pour mysql_stat](#)
- ◇ [Exemple alternative avec mysql_create_db](#)
- ◇ [Exemple au format de date iso 8601:1988](#)
- ◇ [Exemple awt](#)
- ◇ [Exemple com \(1\)](#)
- ◇ [Exemple com \(2\)](#)
- ◇ [Exemple complet avec identification ldap](#)
- ◇ [Exemple complet de vérification de mot de passe avec ldap](#)
- ◇ [Exemple cracklib](#)
- ◇ [Exemple d'affichage d'un message d'erreur oracle après une erreur d'exécution](#)

- ◇ [Exemple d'alternative à mysql_list_fields](#)
- ◇ [Exemple d'alternative à mysql_list_tables](#)
- ◇ [exemple d'envoi de page compressée avec ob_gzhandler](#)
- ◇ [Exemple d'initialisation kadm5](#)
- ◇ [Exemple d'installation avec apache 1.3 sous gentoo](#)
- ◇ [Exemple d'installation de php sous openbsd avec ports](#)
- ◇ [Exemple d'installation sous debian avec apache 1.3](#)
- ◇ [Exemple d'interface](#)
- ◇ [Exemple d'interface de callback readline](#)
- ◇ [Exemple d'introduction](#)
- ◇ [Exemple d'udf sqlite : taille maximale de chaîne](#)
- ◇ [Exemple d'une classe variant, comme utilisé en php 4.x](#)
- ◇ [Exemple d'une classe variant, comme utilisé en php 5](#)
- ◇ [Exemple d'utilisation avec msql](#)
- ◇ [Exemple d'utilisation d'une fonction php dans sqlite](#)
- ◇ [Exemple d'utilisation de max](#)
- ◇ [Exemple d'utilisation de zip](#)
- ◇ [Exemple d'utilisation des nouveaux constructeurs unifiés](#)
- ◇ [Exemple de strstr avec deux arguments](#)
- ◇ [Exemple de swfsprite](#)
- ◇ [Exemple de bufferisation de sortie](#)
- ◇ [Exemple de chaîne heredoc](#)
- ◇ [Exemple de classe abstraite](#)
- ◇ [Exemple de classe finale](#)
- ◇ [Exemple de comparaison d'objets en php 5](#)
- ◇ [Exemple de configuration apache](#)
- ◇ [Exemple de connexion à un serveur ldap ssl](#)
- ◇ [Exemple de connexions d'événements com](#)
- ◇ [Exemple de contrôle de processus](#)
- ◇ [Exemple de création d'un principal](#)
- ◇ [Exemple de duplication d'objets](#)
- ◇ [Exemple de flux qui affiche la page de www.php.net](#)
- ◇ [exemple de fonction d'agrégation sqlite, compatible avec les données binaires](#)
- ◇ [Exemple de fonction de callback](#)
- ◇ [Exemple de gestion d'erreur lors de la création d'image](#)
- ◇ [Exemple de gestion d'erreur lors de la création d'image](#)
- ◇ [Exemple de gestion de sortie avec fonction utilisateur](#)
- ◇ [Exemple de gestion des erreurs durant la création d'image](#)
- ◇ [Exemple de gestion des erreurs durant la création d'une image](#)
- ◇ [Exemple de l'extension zlib](#)
- ◇ [Exemple de lecture de ligne ovrims](#)
- ◇ [Exemple de maître nis](#)
- ◇ [Exemple de message du débogueur](#)
- ◇ [Exemple de modification d'option odbc](#)
- ◇ [Exemple de modification d'un principal](#)
- ◇ [Exemple de modification du mot de passe du principal](#)
- ◇ [Exemple de méthode variable](#)
- ◇ [Exemple de nombres décimaux](#)
- ◇ [Exemple de présentation de l'extension msql](#)
- ◇ [Exemple de retour à l'état initial après l'application d'un patch](#)
- ◇ [Exemple de script avec mbstring](#)
- ◇ [Exemple de session : page2.php](#)
- ◇ [Exemple de surcharge avec __get, __set, __isset et __unset](#)

- ◇ [Exemple de surcharge avec `__call`](#)
- ◇ [Exemple de transaction `payflow_pro`](#)
- ◇ [Exemple de validation `dtl`](#)
- ◇ [Exemple `dotnet`](#)
- ◇ [Exemple `ftp`](#)
- ◇ [Exemple global de l'extension `maxdb`](#)
- ◇ [Exemple général de l'extension `kadm5`](#)
- ◇ [Exemple général de l'extension `pgsql`](#)
- ◇ [Exemple `ipc` avec `socket_create_pair`](#)
- ◇ [Exemple `pdfclock` de la distribution `pdflib 2.0`](#)
- ◇ [Exemple `pdfclock` de `pdflib`](#)
- ◇ [exemple pour `ingres_connect` utilisant le lien par défaut](#)
- ◇ [Exemple pour `ingres_fetch_array`](#)
- ◇ [Exemple pour `ingres_fetch_object`](#)
- ◇ [exemple récursif avec `count` \(php >= 4.2.0\)](#)
- ◇ [Exemple simple avec `fwrite`](#)
- ◇ [Exemple simple avec `recode_file`](#)
- ◇ [Exemple simple avec `recode_string`](#)
- ◇ [Exemple `swf`](#)
- ◇ [Exemples avec `array_slice`](#)
- ◇ [Exemples avec `array_splice`](#)
- ◇ [Exemples avec `array_unshift`](#)
- ◇ [Exemples avec `array_values`](#)
- ◇ [Exemples avec `intval`](#)
- ◇ [Exemples avec `min`](#)
- ◇ [exemples avec `preg_replace_callback` et `create_function`](#)
- ◇ [Exemples avec `print`](#)
- ◇ [Exemples avec `require`](#)
- ◇ [Exemples avec `soundex`](#)
- ◇ [Exemples avec `stripos`](#)
- ◇ [Exemples avec l'opérateur `::`](#)
- ◇ [Exemples avec l'opérateur `parent`](#)
- ◇ [Exemples avec les opérateurs d'assignation](#)
- ◇ [Exemples avec les opérateurs sur les bits](#)
- ◇ [Exemples avec quelques constantes liées au système](#)
- ◇ [Exemples d'interpolation des chaînes de caractères](#)
- ◇ [Exemples de chaînes](#)
- ◇ [Exemples de conversions](#)
- ◇ [Exemples de masques invalides](#)
- ◇ [Exemples de requête](#)
- ◇ [Exemples de typage d'objets](#)
- ◇ [Exemples de vérification de type de variable](#)
- ◇ [Exemples liés à `str_split`](#)
- ◇ [Exemple avec `exif_imagetype`](#)
- ◇ [Exemple avec `exif_read_data`](#)
- ◇ [Exemple avec `exif_thumbnail`](#)
- ◇ [Exemple avec `exit` et un code d'erreur](#)
- ◇ [Exemple avec `exp`](#)
- ◇ [Exporter des classes depuis `php 5.1.0`](#)
- ◇ [Expressions rationnelles](#)
- ◇ [Extension des classes embarquées](#)
- ◇ [Exemple avec `extension_loaded`](#)
- ◇ [Exemple avec `extract`](#)
- ◇ [Extrait du `php.ini`](#)

- ◇ Exécute une requête préparée avec des points d'interrogation comme paramètre fictif
- ◇ Exécute une requête préparée avec un marqueur de positionnement
- ◇ Exécution d'un groupe dans une transaction
- ◇ Exécution d'une commande
- ◇ Exécution d'une commande
- ◇ Exécution d'une requête delete
- ◆ **F**
- ◆
 - ◇ Factorielle avec gmp
 - ◇ Exemple avec fbsql_close
 - ◇ Exemple avec fbsql_connect
 - ◇ Exemple avec fbsql_create_blob
 - ◇ Exemple avec fbsql_create_clob
 - ◇ Exemple avec fbsql_create_clob
 - ◇ Exemple avec fbsql_create_db
 - ◇ Exemple avec fbsql_data_seek
 - ◇ Exemple avec fbsql_erro
 - ◇ Exemple avec fbsql_error
 - ◇ Exemple avec fbsql_fetch_array
 - ◇ Exemple avec fbsql_fetch_assoc
 - ◇ Exemple avec fbsql_fetch_field
 - ◇ Exemple avec fbsql_fetch_object
 - ◇ Exemple avec fbsql_field_name
 - ◇ Exemple avec fbsql_field_type
 - ◇ Exemple avec fbsql_list_dbs
 - ◇ Exemple avec fbsql_list_fields
 - ◇ Exemple avec fbsql_next_result
 - ◇ Exemple avec fbsql_query (2)
 - ◇ Exemple avec fbsql_read_blob
 - ◇ Exemple avec fbsql_read_clob
 - ◇ Exemple avec fbsql_tablename
 - ◇ Exemple avec fclose
 - ◇ Exemple avec feof et un pointeur de fichier invalide
 - ◇ Ferme un fichier de base de données dbase
 - ◇ Fermer une connexion informix
 - ◇ Fermeture d'un bloc de mémoire partagée
 - ◇ Fermeture d'une ressource de fichier
 - ◇ Exemple avec fgetc
 - ◇ Fichier utilisateurs.txt
 - ◇ Fichier batch pour exécuter un script php en ligne de commande (script.bat)
 - ◇ Fichier d'inclusion exemple.inc pour la chaîne xml d'exemple
 - ◇ Exemple avec file
 - ◇ Exemple avec fileatime
 - ◇ Exemple avec filectime
 - ◇ Exemple avec filemtime
 - ◇ Exemple avec filesize
 - ◇ Exemple avec filetype
 - ◇ Filtrage dynamique d'un flux
 - ◇ Exemple avec floatval
 - ◇ Exemple avec flock
 - ◇ Exemple avec floor
 - ◇ Exemple avec fmod
 - ◇ Fonction php qui scanne les titres sur un serveur yaz
 - ◇ Fonction singleton

- ◇ [Fonctions calendrier](#)
- ◇ [Fonctions d'objets : test_script.php](#)
- ◇ [Fonctions récursives](#)
- ◇ [Fonctions à nombre d'arguments variable](#)
- ◇ [Exemple avec fopen sous windows](#)
- ◇ [Foreach en php 5](#)
- ◇ [Format des questionnaires avec xslt_set_sax_handler](#)
- ◇ [Format monétaire avec fprintf](#)
- ◇ [Formatage avec date](#)
- ◇ [Exemple avec fputs](#)
- ◇ [Exemple avec fseek](#)
- ◇ [Exemple avec fstat](#)
- ◇ [Exemple avec ftell](#)
- ◇ [Exemple avec ftp_alloc](#)
- ◇ [Exemple avec ftp_cdup](#)
- ◇ [Exemple avec ftp_chdir](#)
- ◇ [Exemple avec ftp_chmod](#)
- ◇ [Exemple avec ftp_close](#)
- ◇ [Exemple avec ftp_connect](#)
- ◇ [Exemple avec ftp_delete](#)
- ◇ [Exemple avec ftp_exec](#)
- ◇ [Exemple avec ftp_fget](#)
- ◇ [Exemple avec ftp_fput](#)
- ◇ [Exemple avec ftp_get](#)
- ◇ [Exemple avec ftp_get_option](#)
- ◇ [Exemple avec ftp_login](#)
- ◇ [Exemple avec ftp_mdtm](#)
- ◇ [Exemple avec ftp_mkdir](#)
- ◇ [ftp_nb_continue exemple](#)
- ◇ [Exemple avec ftp_nb_fget](#)
- ◇ [Exemple avec ftp_nb_fput](#)
- ◇ [Exemple avec ftp_nlist](#)
- ◇ [Exemple avec ftp_pasv](#)
- ◇ [Exemple avec ftp_put](#)
- ◇ [Exemple avec ftp_pwd](#)
- ◇ [Exemple avec ftp_rawlist](#)
- ◇ [Exemple avec ftp_rename](#)
- ◇ [Exemple avec ftp_rmdir](#)
- ◇ [Exemple avec ftp_set_option](#)
- ◇ [Exemple avec ftp_size](#)
- ◇ [Exemple avec ftp_ssl_connect](#)
- ◇ [Exemple avec ftp_systype](#)
- ◇ [Exemple avec func_get_arg](#)
- ◇ [Exemple avec func_get_args](#)
- ◇ [Exemple avec func_num_args](#)
- ◇ [Exemple avec function_exists](#)
- ◆ **G**
- ◆ [Générer et intercepter une erreur](#)
- ◇ [Gestion d'erreurs avancées en php](#)
- ◇ [gestion d'erreurs avec xslt_error et xslt_erno](#)
- ◇ [Gestion des assertions avec un gestionnaire personnalisé](#)
- ◇ [Gestion des erreurs par exception avec soap](#)
- ◇ [Gestion des messages non traités de sybase_set_message_handler](#)
- ◇ [Gestionnaire d'erreurs avec set_error_handler et trigger_error](#)

- ◇ [Gestionnaire d'exceptions java](#)
- ◇ [Gestionnaire orienté objet](#)
- ◇ [Exemple avec get_browser : liste des capacités du navigateur du client](#)
- ◇ [Exemple avec get_class](#)
- ◇ [Exemple avec get_class_methods](#)
- ◇ [Exemple avec get_class_vars](#)
- ◇ [Exemple avec get_declared_classes](#)
- ◇ [Exemple avec get_declared_interfaces](#)
- ◇ [Exemple avec get_defined_constants](#)
- ◇ [Exemple avec get_defined_functions](#)
- ◇ [Exemple avec get_defined_vars](#)
- ◇ [Exemple avec get_extension_funcs](#)
- ◇ [Exemple avec get_headers](#)
- ◇ [Exemple avec get_include_path](#)
- ◇ [Exemple avec get_included_files \(abc.php \)](#)
- ◇ [Exemple avec get_loaded_extensions](#)
- ◇ [Exemple avec get_magic_quotes_gpc](#)
- ◇ [Exemple avec get_object_vars](#)
- ◇ [Exemple avec get_parent_class](#)
- ◇ [Exemple avec get_resource_type](#)
- ◇ [Exemple avec getcwd](#)
- ◇ [Exemple avec getdate](#)
- ◇ [Exemple avec getenv](#)
- ◇ [Exemple avec gethostbyaddr](#)
- ◇ [Exemple avec gethostbyaddr](#)
- ◇ [Exemple avec gethostbyname](#)
- ◇ [getimagesize qui retourne iptc](#)
- ◇ [Exemple avec getlastmod](#)
- ◇ [Exemple avec getopt](#)
- ◇ [Exemple avec getprotobyname](#)
- ◇ [Exemple avec getrusage](#)
- ◇ [Exemple avec getservbyname](#)
- ◇ [Exemple avec gettimeofday](#)
- ◇ [Exemple avec gmdate](#)
- ◇ [Exemple avec gmmktime à la frontière](#)
- ◇ [Exemple avec gmp_abs](#)
- ◇ [Exemple avec gmp_add](#)
- ◇ [Exemple avec gmp_and](#)
- ◇ [Exemple avec gmp_clrbit](#)
- ◇ [Exemple avec gmp_cmp](#)
- ◇ [Exemple avec gmp_com](#)
- ◇ [Exemple avec gmp_div_q](#)
- ◇ [Exemple avec gmp_div_r](#)
- ◇ [Exemple avec gmp_divexact](#)
- ◇ [Exemple avec gmp_fact](#)
- ◇ [Exemple avec gmp_gcd](#)
- ◇ [Exemple avec gmp_hamdist](#)
- ◇ [Exemple avec gmp_intval](#)
- ◇ [Exemple avec gmp_invert](#)
- ◇ [Exemple avec gmp_jacobi](#)
- ◇ [Exemple avec gmp_legendre](#)
- ◇ [Exemple avec gmp_mod](#)
- ◇ [Exemple avec gmp_mul](#)
- ◇ [Exemple avec gmp_neg](#)

- ◇ [Exemple avec gmp_or](#)
- ◇ [Exemple avec gmp_perfect_square](#)
- ◇ [Exemple avec gmp_popcount](#)
- ◇ [Exemple avec gmp_pow](#)
- ◇ [Exemple avec gmp_powm](#)
- ◇ [Exemple avec gmp_prob_prime](#)
- ◇ [Exemple avec gmp_random](#)
- ◇ [Exemple avec gmp_scan0](#)
- ◇ [Exemple avec gmp_scan1](#)
- ◇ [Exemple avec gmp_setbit](#)
- ◇ [Exemple avec gmp_sign](#)
- ◇ [Exemple avec gmp_sqrt](#)
- ◇ [Exemple avec gmp_sqrtrem](#)
- ◇ [Exemple avec gmp_sub](#)
- ◇ [Exemple avec gmp_xor](#)
- ◇ [Exemple avec gmstrftime](#)
- ◇ [Grouper toutes les valeurs d'une seule colonne](#)
- ◇ [Exemple avec gzclose](#)
- ◇ [Exemple avec gzcompress](#)
- ◇ [Exemple avec gzdeflate](#)
- ◇ [Exemple avec gzeof](#)
- ◇ [Exemple avec gzfile](#)
- ◇ [Exemple avec gzgetc](#)
- ◇ [Exemple avec gzgets](#)
- ◇ [Exemple avec gzgetss](#)
- ◇ [Exemple avec gzinflate](#)
- ◇ [Exemple avec gzopen](#)
- ◇ [Exemple avec gzpassthru](#)
- ◇ [Exemple avec gzread](#)
- ◇ [Exemple avec gzseek](#)
- ◇ [Exemple avec gzuncompress](#)
- ◇ [Exemple avec gzwrite](#)
- ◇ [Génération d'un fichier de "traces"](#)
- ◆ **H**
- ◆ [Exemple avec headers_list](#)
- ◆ [Exemple avec headers_sent](#)
- ◆ [Exemple avec hexdec](#)
- ◆ [Exemple avec highlight_string](#)
- ◆ [Exemple avec htmlentities](#)
- ◆ [Exemple avec htmlspecialchars](#)
- ◆ [Exemple avec htmlspecialchars_decode](#)
- ◆ [Http:// et redirections](#)
- ◆ [Httpd.conf fournit par apache](#)
- ◆ [Héritage d'une classe simple](#)
- ◆ [Héritage de classes \(2\)](#)
- ◆ **I**
- ◆ [Exemple avec ibase_blob_get](#)
- ◆ [Exemple avec ibase_blob_import](#)
- ◆ [Exemple avec ibase_execute](#)
- ◆ [Exemple avec ibase_fetch_object](#)
- ◆ [Exemple avec ibase_field_info](#)
- ◆ [Exemple avec ibase_name_result](#)
- ◆ [Exemple avec ibase_num_fields](#)
- ◆ [Exemple avec ibase_query](#)

- ◇ [Exemple avec ibase_set_event_handler](#)
- ◇ [Exemple avec ibase_timefmt](#)
- ◇ [Exemple avec iconv](#)
- ◇ [Exemple avec iconv_get_encoding](#)
- ◇ [Exemple avec iconv_mime_decode](#)
- ◇ [Exemple avec iconv_mime_decode_headers](#)
- ◇ [Exemple avec iconv_mime_encode](#)
- ◇ [Exemple avec iconv_set_encoding](#)
- ◇ [Exemple avec id3_get_frame_long_name](#)
- ◇ [Exemple avec id3_get_frame_short_name](#)
- ◇ [Exemple avec id3_get_genre_id](#)
- ◇ [Exemple avec id3_get_genre_list](#)
- ◇ [Exemple avec id3_get_genre_name](#)
- ◇ [Exemple avec id3_get_tag](#)
- ◇ [Exemple avec id3_get_version](#)
- ◇ [Exemple avec id3_remove_tag](#)
- ◇ [Exemple avec id3_set_tag](#)
- ◇ [Exemple avec idate avec comme retour, un entier d'un seul chiffre](#)
- ◇ [Identification avec un mot de passe](#)
- ◇ [Identification en utilisant une clé d'hôte publique](#)
- ◇ [Identification en utilisant une clé publique](#)
- ◇ [Identification http avec nom d'utilisateur/mot de passe forcé](#)
- ◇ [Exemple avec ifx_errormsg](#)
- ◇ [Exemple avec ifx_fetch_row](#)
- ◇ [Exemple avec ifx_fieldproperties](#)
- ◇ [Exemple avec image2wbmp](#)
- ◇ [Image type to mime type \(fichier\)](#)
- ◇ [Exemple avec imagechar](#)
- ◇ [Exemple avec imagecharup](#)
- ◇ [Exemple avec imagecolorallocate](#)
- ◇ [Exemple avec imagecolorallocatealpha](#)
- ◇ [Exemple avec imagecolordeallocate](#)
- ◇ [Exemple avec imagecolorsforindex](#)
- ◇ [Exemple avec imagecreatefromstring](#)
- ◇ [Exemple avec imageellipse](#)
- ◇ [Exemple avec imagefilledellipse](#)
- ◇ [Exemple avec imagefilledpolygon](#)
- ◇ [Exemple avec imagefilter en utilisant le paramètre img_filter_colorize](#)
- ◇ [Exemple avec imageloadfont](#)
- ◇ [Exemple avec imagepng](#)
- ◇ [Exemple avec imagepolygon](#)
- ◇ [Exemple avec imagesloadfont](#)
- ◇ [Exemple avec imagesetstyle](#)
- ◇ [Exemple avec imagestring](#)
- ◇ [Exemple avec imagesx](#)
- ◇ [Exemple avec imagesy](#)
- ◇ [Exemple avec imagettftext](#)
- ◇ [Exemple avec imagetypes](#)
- ◇ [Exemple avec imap_append](#)
- ◇ [Exemple avec imap_check](#)
- ◇ [Exemple avec imap_createmailbox](#)
- ◇ [Exemple avec imap_delete](#)
- ◇ [Exemple avec imap_fetch_overview](#)
- ◇ [Exemple avec imap_get_quota pour php 4.3](#)

- ◇ [Exemple avec imap_get_quotaroot](#)
- ◇ [Exemple avec imap_getmailboxes](#)
- ◇ [Exemple avec imap_list](#)
- ◇ [Exemple avec imap_mail_compose](#)
- ◇ [Exemple avec imap_mailboxmsginfo](#)
- ◇ [Exemple avec imap_open](#)
- ◇ [Exemple avec imap_ping](#)
- ◇ [Exemple avec imap_rfc822_parse_adrlist](#)
- ◇ [Exemple avec imap_rfc822_write_address](#)
- ◇ [Exemple avec imap_set_quota](#)
- ◇ [Exemple avec imap_setflag_full](#)
- ◇ [Exemple avec imap_status](#)
- ◇ [Exemple avec imap_thread](#)
- ◇ [Exemple avec implode](#)
- ◇ [Import dom](#)
- ◇ [Exemple avec import_request_variables](#)
- ◇ [Importation d'un objet simplexml dans dom avec dom_import_simplexml](#)
- ◇ [exemple avec in_array et deux tableaux en paramètres](#)
- ◇ [include_once est insensible à la casse sous windows](#)
- ◇ [Include_path sous windows](#)
- ◇ [Inclure une image php en pdf](#)
- ◇ [Indexation automatique de tableau](#)
- ◇ [Exemple avec inet_ntop](#)
- ◇ [Exemple avec inet_pton](#)
- ◇ [Exemple avec ingres_cursor](#)
- ◇ [Exemple avec ingres_errno](#)
- ◇ [Exemple avec ingres_error](#)
- ◇ [Exemple avec ingres_errsqlstate](#)
- ◇ [Exemple avec ingres_fetch_row](#)
- ◇ [Exemple avec ingres_query](#)
- ◇ [Exemple avec ini_get_all](#)
- ◇ [initialisation d'une nouvelle session curl et recherche d'une page web](#)
- ◇ [initialise une session curl et récupère une page web](#)
- ◇ [initialise une session curl et récupère une page web](#)
- ◇ [initialiser une session curl et récupérer une page web](#)
- ◇ [Insertion d'un enregistrement dans une base de données dbase](#)
- ◇ [Insertion d'une image dans une base de données oracle](#)
- ◇ [Insertion de valeurs dans la table "catalogue" ifx](#)
- ◇ [Installation de php avec les flags use](#)
- ◇ [Installation de rar](#)
- ◇ [Installation de tidy à la main pour php 4.3.x](#)
- ◇ [Installation de yaz pecl sur linux gnu/debian](#)
- ◇ [Installation des modules php de mysql, curl et gd](#)
- ◇ [Installation manuelle de l'extension xattr](#)
- ◇ [Installation à la main de tcpwrap](#)
- ◇ [Installation à la main de xdiff](#)
- ◇ [Instruction directive en contexte global](#)
- ◇ [Instruction elseif](#)
- ◇ [Instruction for alternative](#)
- ◇ [Instruction foreach et while \(3\)](#)
- ◇ [Instruction if \(\) else](#)
- ◇ [Instruction switch alternative](#)
- ◇ [Instruction while et break](#)
- ◇ [instruction alternative if\(\) elseif\(\) et else](#)

- ◇ [Instruction break](#)
- ◇ [Instruction d'installation \(module partagé apache 2\)](#)
- ◇ [Instruction if \(\) et bloc](#)
- ◇ [instructions d'installation \(installation en tant que module statique d'apache\) de php](#)
- ◇ [Instructions d'installation caudium](#)
- ◇ [instructions d'installation de php \(en module apache\)](#)
- ◇ [Exemple avec interface_exists](#)
- ◇ [Interopérabilité dom](#)
- ◇ [Interprétation d'un document xml](#)
- ◇ [Interprétation d'une chaîne xml](#)
- ◇ [Interprétation des nombres octaux](#)
- ◇ [Introduction à la mémoire partagée](#)
- ◇ [Inverser une chaîne avec strrev](#)
- ◇ [Exemple avec ircg_fetch_error_msg](#)
- ◇ [Exemple avec ircg_list](#)
- ◇ [Exemple avec is_array](#)
- ◇ [Exemple avec is_bool](#)
- ◇ [Exemple avec is_callable](#)
- ◇ [Exemple avec is_dir](#)
- ◇ [Exemple avec is_executable](#)
- ◇ [Exemple avec is_file](#)
- ◇ [Exemple avec is_readable](#)
- ◇ [Exemple avec is_resource](#)
- ◇ [Exemple avec is_scalar](#)
- ◇ [Exemple avec is_subclass_of](#)
- ◇ [Exemple avec is_uploaded_file en php 4 < 4.0.3.](#)
- ◇ [Exemple avec is_writable](#)
- ◇ [Exemple avec isset et un tableau](#)
- ◇ [Itération d'un objet implémentant iteratoraggregate](#)
- ◆ **J**
- ◆ [Exemple avec jdtojewish](#)
- ◆ [Jeux de caractères qui risquent de ne pas fonctionner en php](#)
- ◆ **K**
- ◆ [Exemple avec kadm5_delete_principal](#)
- ◆ [Exemple avec kadm5_get_policies](#)
- ◆ [Exemple avec kadm5_get_principal](#)
- ◆ [Exemple avec kadm5_get_principals](#)
- ◆ [Exemple avec key](#)
- ◆ [Exemple avec krsort](#)
- ◆ [Exemple avec ksort](#)
- ◆ **L**
- ◆ [Exemple avec l'extension memcache](#)
- ◆ [Exemple avec l'extension rar](#)
- ◆ [la classe d'un objet doit être définie avant sa délinéarisation](#)
- ◆ [La classe pdoexception](#)
- ◆ [La génération automatique de clé peut utiliser un ancien maximum](#)
- ◆ [Exemple avec la table de traduction des caractères en entités html](#)
- ◆ [La valeur null](#)
- ◆ [Lancer une exception](#)
- ◆ [Exemple avec le domaine par défaut](#)
- ◆ [Exemple avec le filtre string.strip_tags](#)
- ◆ [Exemple avec le paramètre autoload](#)
- ◆ [Exemple avec le paramètre convmap](#)

- ◇ [Exemple avec le paramètre limit](#)
- ◇ [Lecture d'un fichier chiffré](#)
- ◇ [Lecture d'un fichier ligne par ligne](#)
- ◇ [Lecture d'un lien symbolique](#)
- ◇ [Lecture de tous les enregistrements de la base de données](#)
- ◇ [les arguments à valeur par défaut doivent être en premier : valide](#)
- ◇ [Les commentaires de type c](#)
- ◇ [Les deux lignes relatives à dotdeb](#)
- ◇ [Les directives liées aux tailles mémoires sont lues littéralement](#)
- ◇ [exemple avec les filtres convert.quoted-printable-encode et convert.quoted-printable-decode](#)
- ◇ [Exemple avec les paramètres optionnels](#)
- ◇ [Les références ne sont pas des pointeurs](#)
- ◇ [Exemple avec les sockets : client tcp/ip simple](#)
- ◇ [Les variables sont locales à la fonction \(2\)](#)
- ◇ [Les variables statiques et les références \(2\)](#)
- ◇ [Les variables super globales](#)
- ◇ [Exemple avec levenshtein](#)
- ◇ [Exemple avec libxml_get_errors](#)
- ◇ [Exemple avec libxml_set_streams_context](#)
- ◇ [Exemple avec libxml_use_internal_errors](#)
- ◇ [Lie l'affichage du jeu de résultats à des variables php](#)
- ◇ [Ligne d'invocation de php](#)
- ◇ [Exemple avec linkinfo](#)
- ◇ [Lire des informations dbm depuis la ligne de commande](#)
- ◇ [Lire des messages d'erreurs avec popen](#)
- ◇ [Lire l'identifiant public avec dom xml](#)
- ◇ [Lire la taille d'un bloc de mémoire partagée](#)
- ◇ [Lire les valeurs de sqlca.sqlerrd\[x\]](#)
- ◇ [Lire tous les objet avec snmpwalkoid \(2\)](#)
- ◇ [Lire un bloc de mémoire partagée](#)
- ◇ [Lire un document fdf sous forme de chaîne](#)
- ◇ [Lire un identifiant système dom xml](#)
- ◇ [Lire un nom de domaine dans une url](#)
- ◇ [Exemple avec list en tenant compte de l'ordre](#)
- ◇ [Listage des noeuds fils](#)
- ◇ [liste de tous les fichiers dans le répertoire courant et enlève les . et ..](#)
- ◇ [Liste de tous les membres enregistrés dans la base de données](#)
- ◇ [Liste des clés autorisées avec ssh2_publickey_list](#)
- ◇ [Liste les noeuds sous dom xml](#)
- ◇ [liste toutes les lignes de table "phone" sous forme de table html](#)
- ◇ [Liste toutes les valeurs de l'attribut "mail" d'une entrée](#)
- ◇ [lit et affiche le contenu d'un fichier csv avec fgetcsv](#)
- ◇ [Lit la durée de vie de la machine, et l'affiche](#)
- ◇ [Lit un exemple](#)
- ◇ [Lit un objet hyperwave et le vérifie](#)
- ◇ [Exemple avec localeconv](#)
- ◇ [Exemple avec ltrim](#)
- ◆ **M**
- ◆ [Manipuler des éléments de tableau](#)
- ◇ [Exemple avec maxdb_connect_errno](#)
- ◇ [Exemple avec maxdb_connect_error](#)
- ◇ [Exemple avec maxdb_get_client_info](#)
- ◇ [Exemple avec maxdb_get_client_version](#)

- ◇ [Exemple avec mb_convert_encoding](#)
- ◇ [Exemple avec mb_convert_kana](#)
- ◇ [Exemple avec mb_convert_kana](#)
- ◇ [Exemple avec mb_convert_variables](#)
- ◇ [Exemple avec mb_detect_encoding](#)
- ◇ [Exemple avec mb_detect_order](#)
- ◇ [Exemple avec mb_encode_numericentity](#)
- ◇ [Exemple avec mb_internal_encoding](#)
- ◇ [Exemple avec mb_list_encodings](#)
- ◇ [Exemple avec mb_output_handler](#)
- ◇ [Exemple avec mb_preferred_mime_string](#)
- ◇ [Exemple avec mb_striwidth](#)
- ◇ [Exemple avec mb_strtolower](#)
- ◇ [Exemple avec mb_strtoupper](#)
- ◇ [Exemple avec mb_substitute_character](#)
- ◇ [Exemple avec mb_substr_count](#)
- ◇ [Exemple avec mcrypt_create_iv](#)
- ◇ [Exemple avec mcrypt_enc_get_algorithms_name](#)
- ◇ [Exemple avec mcrypt_enc_get_modes_name](#)
- ◇ [Exemple avec mcrypt_enc_get_supported_key_sizes](#)
- ◇ [Exemple avec mcrypt_encrypt](#)
- ◇ [Exemple avec mcrypt_get_block_size](#)
- ◇ [Exemple avec mcrypt_get_block_size](#)
- ◇ [Exemple avec mcrypt_get_block_size](#)
- ◇ [Exemple avec mcrypt_get_cipher_name](#)
- ◇ [Exemple avec mcrypt_list_algorithms](#)
- ◇ [Exemple avec mcrypt_list_modes](#)
- ◇ [Exemple avec mcrypt_module_self_test](#)
- ◇ [Exemple avec md5](#)
- ◇ [Exemple avec mdcrypt_generic](#)
- ◇ [Meilleure pratique](#)
- ◇ [Exemple avec memcache::add](#)
- ◇ [Exemple avec memcache::close](#)
- ◇ [Exemple avec memcache::connect](#)
- ◇ [Exemple avec memcache::decrement](#)
- ◇ [Exemple avec memcache::delete](#)
- ◇ [Exemple avec memcache::flush](#)
- ◇ [Exemple avec memcache::get](#)
- ◇ [Exemple avec memcache::getversion](#)
- ◇ [Exemple avec memcache::increment](#)
- ◇ [Exemple avec memcache::pconnect](#)
- ◇ [Exemple avec memcache::replace](#)
- ◇ [Exemple avec memcache::set](#)
- ◇ [Exemple avec memory_get_usage](#)
- ◇ [Exemple avec method_exists](#)
- ◇ [Exemple avec mhash_get_hash_name](#)
- ◇ [Exemple avec microtime en php 5](#)
- ◇ [Migration : nouvelle syntaxe if...endif](#)
- ◇ [Migration : nouvelle syntaxe while...endwhile](#)
- ◇ [Migration depuis 2.0 : concaténation de chaînes](#)
- ◇ [Migration depuis 2.0 : valeur retournées, nouvelle façon](#)
- ◇ [Migration des fichiers de configuration apache, en mode cgi](#)
- ◇ [Migration des variables globales](#)
- ◇ [Mise en gras d'un mot dans un texte](#)

- ◇ [Mise en évidence des résultats de la recherche](#)
- ◇ [Mise à jour d'un enregistrement dans une base de données](#)
- ◇ [Exemple avec mkdir](#)
- ◇ [Modification d'une variable d'environnement](#)
- ◇ [Modification d'une variable d'environnement d'apache avec apache_setenv](#)
- ◇ [Modification des contrôles du serveur](#)
- ◇ [Modification du propriétaire avant la transformation](#)
- ◇ [Modifier les paramètres de configuration sesam](#)
- ◇ [Exemple avec money_format](#)
- ◇ [Exemple avec mysql_fetch_array](#)
- ◇ [Exemple avec mysql_fetch_object](#)
- ◇ [Exemple avec mysql_fetch_row](#)
- ◇ [Exemple avec mysql_query](#)
- ◇ [Exemple avec mssql_next_result](#)
- ◇ [Exemple avec mssql_select_db](#)
- ◇ [Exemple avec mt_rand](#)
- ◇ [Exemple avec mt_srand](#)
- ◇ [Exemple avec multicolor.php">swfdisplayitem->multicolor](#)
- ◇ [Exemple avec mysql_affected_rows en utilisant les transactions](#)
- ◇ [Exemple avec mysql_client_encoding](#)
- ◇ [Exemple avec mysql_close](#)
- ◇ [Exemple avec mysql_connect en utilisant la syntaxe ":/path/to/socket"](#)
- ◇ [Exemple avec mysql_data_seek](#)
- ◇ [Exemple avec mysql_drop_db](#)
- ◇ [Exemple avec mysql_errno](#)
- ◇ [Exemple avec mysql_error](#)
- ◇ [Exemple avec mysql_escape_string](#)
- ◇ [mysql_fetch_array avec mysql_both](#)
- ◇ [Exemple avec mysql_fetch_assoc](#)
- ◇ [Exemple avec mysql_fetch_field](#)
- ◇ [Exemple avec mysql_fetch_lengths](#)
- ◇ [Exemple avec mysql_fetch_object](#)
- ◇ [Exemple avec mysql_fetch_row](#)
- ◇ [Exemple avec mysql_field_flags](#)
- ◇ [Exemple avec mysql_field_len](#)
- ◇ [Exemple avec mysql_field_name](#)
- ◇ [Exemple avec mysql_field_table](#)
- ◇ [Exemple avec mysql_field_type](#)
- ◇ [Exemple avec mysql_free_result](#)
- ◇ [Exemple avec mysql_get_client_info](#)
- ◇ [Exemple avec mysql_get_host_info](#)
- ◇ [Exemple avec mysql_get_proto_info](#)
- ◇ [Exemple avec mysql_get_server_info](#)
- ◇ [Exemple avec mysql_insert_id](#)
- ◇ [Exemple avec mysql_list_dbs](#)
- ◇ [Exemple avec mysql_list_processes](#)
- ◇ [Exemple avec mysql_num_fields](#)
- ◇ [Exemple avec mysql_num_rows](#)
- ◇ [Exemple avec mysql_ping](#)
- ◇ [Exemple avec mysql_result](#)
- ◇ [Exemple avec mysql_select_db](#)
- ◇ [Exemple avec mysql_tablename](#)
- ◇ [Exemple avec mysql_thread_id](#)
- ◇ [Exemple avec mysqli_connect_errno](#)

- ◇ [Exemple avec mysqli_connect_error](#)
- ◇ [Exemple avec mysqli_get_client_info](#)
- ◇ [Exemple avec mysqli_get_client_version](#)
- ◇ [Exemple avec mysqli_report](#)
- ◇ [Méthode de paramètre d'usine](#)
- ◆ **N**
- ◆ [N'oubliez pas le point-virgule après continue](#)
- ◆ [N'utiliser pas com_get\(\), utilisez la syntaxe oo à la place](#)
- ◆ [N'utiliser pas com_invoke\(\), utilisez la syntaxe oo à la place](#)
- ◆ [N'utiliser pas com_load\(\), utilisez la syntaxe oo à la place](#)
- ◆ [N'utiliser pas com_set\(\), utilisez la syntaxe oo à la place](#)
- ◆ [Exemple avec natcasesort](#)
- ◆ [Exemple avec natsort](#)
- ◆ [Exemple avec ncurses_getmouse](#)
- ◆ [Exemple avec ncurses_mousemask](#)
- ◆ [Exemple avec next et ses amies](#)
- ◆ [Exemple avec ngettext](#)
- ◆ [Exemple avec nl2br](#)
- ◆ [Nom de champs et type sql.](#)
- ◆ [Nom du catalogue de messages sesam](#)
- ◆ [Nombre de lignes affectées](#)
- ◆ [Nombre variable d'argument sous forme de tableau](#)
- ◆ [Noms valides et invalides pour les constantes](#)
- ◆ [Nouveau comportement de strtok](#)
- ◆ [Nouvelle syntaxe php3](#)
- ◆ [Nouvelles balises php](#)
- ◆ [Exemple avec nsapi_request_headers](#)
- ◆ [Exemple avec number_format](#)
- ◆ **O**
- ◆ [Exemple avec ob_end_clean](#)
- ◆ [Exemple avec ob_end_flush](#)
- ◆ [Exemple avec ob_get_clean](#)
- ◆ [Exemple avec ob_get_flush](#)
- ◆ [Exemple avec ob_iconv_handler](#)
- ◆ [Exemple avec ob_list_handlers](#)
- ◆ [Exemple avec ob_tidyhandler](#)
- ◆ [Obtenir la liste des paquets supplémentaires](#)
- ◆ [Obtenir un contenu](#)
- ◆ [oci_bind_by_name et les caractères d'espace terminaux \(2\)](#)
- ◆ [Exemple avec oci_commit](#)
- ◆ [Exemple avec oci_connect](#)
- ◆ [Exemple avec oci_define_by_name](#)
- ◆ [Exemple avec oci_fetch_all](#)
- ◆ [Exemple avec oci_fetch_array et oci_return_lob](#)
- ◆ [Exemple avec oci_field_name](#)
- ◆ [Exemple avec oci_field_size](#)
- ◆ [Exemple avec oci_field_type](#)
- ◆ [Exemple avec oci_new_connect](#)
- ◆ [Exemple avec oci_new_descriptor](#)
- ◆ [Exemple avec oci_num_fields](#)
- ◆ [Exemple avec oci_num_rows](#)
- ◆ [Exemple avec oci_server_version](#)
- ◆ [Exemple avec ocifetchinto](#)
- ◆ [Exemple avec ocistatementtype](#)

- ◇ [Exemple avec octdec](#)
- ◇ [Exemple avec odbc_fetch_into en php 4.2.0](#)
- ◇ [Exemple avec odbc_result](#)
- ◇ [Exemple avec opendir](#)
- ◇ [Exemple avec openssl_csr_new - création d'un certificat auto chiffré](#)
- ◇ [Exemple avec openssl_csr_sign : signer une csr \(comment réaliser votre propre autorité de certification\)](#)
- ◇ [Exemple avec openssl_error_string](#)
- ◇ [Exemple avec openssl_open](#)
- ◇ [Exemple avec openssl_pkcs7_decrypt](#)
- ◇ [Exemple avec openssl_pkcs7_encrypt](#)
- ◇ [Exemple avec openssl_pkcs7_sign](#)
- ◇ [Exemple avec openssl_seal](#)
- ◇ [Exemple avec openssl_sign](#)
- ◇ [Exemple avec openssl_verify](#)
- ◇ [Opérateur d'exécution](#)
- ◇ [Opérateur de concaténation](#)
- ◇ [Opérateur de contrôle d'erreur](#)
- ◇ [Opérateur de type : instanceof](#)
- ◇ [Opérations arithmétiques sur un caractère](#)
- ◇ [Exemple avec ora_bind](#)
- ◇ [Exemple avec ora_fetch_into](#)
- ◇ [Exemple avec ora_logon](#)
- ◇ [Exemple avec ord](#)
- ◇ [Exemple avec output_add_rewrite_var](#)
- ◇ [Exemple avec output_reset_rewrite_vars](#)
- ◇ [Ouverture d'un fichier](#)
- ◇ [Ouverture d'un fichier de base de donnéesibase](#)
- ◇ [Ouverture d'un fichier via sftp](#)
- ◇ [Ouverture d'un flux depuis une connexion active](#)
- ◇ [Ouverture d'un shell et récupération du flux stderr qui lui est associé.](#)
- ◇ [Ouverture d'un tunnel sur un hôte arbitraire](#)
- ◇ [Ouverture d'une base de données paradox](#)
- ◇ [ouverture d'une connexion forçant 3des-cbc lors de l'envoi de paquets, n'importe quel force aes cipher lors de la réception de paquets, aucune compression dans les deux directions, et change de clés group1.](#)
- ◇ [Ouvrir un document xml à partir d'un fichier](#)
- ◇ [Ouvrir un document xml à partir d'une chaîne](#)
- ◇ [Overload avec une classe php](#)
- ◇ [Exemple avec override_function](#)
- ◇ [Exemple avec ovrimos_connect](#)
- ◇ [Exemple avec ovrimos_result_all](#)
- ◆ P
- ◆
 - ◇ [Parcourir la liste des hash](#)
 - ◇ [Parcours de toutes les entrées de la table](#)
 - ◇ [Exemple avec parse_ini_file](#)
 - ◇ [Exemple avec parse_str](#)
 - ◇ [Exemple avec parse_url](#)
 - ◇ [Exemple avec parsekit_compile_file](#)
 - ◇ [Exemple avec parsekit_compile_string](#)
 - ◇ [Exemple avec parsekit_func_arginfo](#)
 - ◇ [parsemolddb.php : analyse molddb.xml et crée un tableau d'objets moléculaires](#)
 - ◇ [Passage d'arguments aux fonctions sandbox](#)
 - ◇ [Passage d'arguments par référence](#)

- ◇ [Passage de paramètre par références](#)
- ◇ [Passage par référence invalides](#)
- ◇ [Passer des données fdf à un second formulaire](#)
- ◇ [Passer des variables php à un fichier xls](#)
- ◇ [Passer du mode php au mode html et vice-versa](#)
- ◇ [Passer en revue un tableau avec each](#)
- ◇ [Passer en revue une base](#)
- ◇ [Passer en revue une base de données d'une base dbm](#)
- ◇ [Exemple avec pathinfo](#)
- ◇ [Exemple avec pcntl_fork](#)
- ◇ [Exemple avec pcntl_signal](#)
- ◇ [Exemple avec pdf_add_bookmark](#)
- ◇ [Exemple avec pdf_arcn](#)
- ◇ [Exemple avec pdf_arcn](#)
- ◇ [Exemple avec pdf_circle](#)
- ◇ [Exemple avec pdf_findfont](#)
- ◇ [Exemple avec pdf_setcolor : couleurs cmyk](#)
- ◇ [Exemple avec pdo::getavailabledrivers](#)
- ◇ [Exemple avec pdo::query](#)
- ◇ [Exemple avec pdostatement::closecursor](#)
- ◇ [Personnalisation de la page de crédits php](#)
- ◇ [Petit exemple avec bzip2](#)
- ◇ [Exemple avec pfprow_process_raw](#)
- ◇ [pg_affected_rows example](#)
- ◇ [Exemple avec pg_cancel_query](#)
- ◇ [Exemple avec pg_client_encoding](#)
- ◇ [Exemple avec pg_close](#)
- ◇ [Exemple avec pg_connect](#)
- ◇ [Exemple avec pg_connection_busy](#)
- ◇ [Exemple avec pg_connection_reset](#)
- ◇ [Exemple avec pg_connection_status](#)
- ◇ [Exemple avec pg_convert](#)
- ◇ [Exemple avec pg_copy_from](#)
- ◇ [Exemple avec pg_copy_to](#)
- ◇ [Exemple avec pg_dbname](#)
- ◇ [Exemple avec pg_delete](#)
- ◇ [Exemple avec pg_end_copy](#)
- ◇ [Exemple avec pg_escape_bytea](#)
- ◇ [Exemple avec pg_escape_string](#)
- ◇ [Exemple avec pg_execute](#)
- ◇ [Exemple avec pg_fetch_all](#)
- ◇ [Exemple avec pg_fetch_all_columns](#)
- ◇ [Exemple avec pg_fetch_array](#)
- ◇ [Exemple avec pg_fetch_assoc](#)
- ◇ [Exemple avec pg_fetch_object](#)
- ◇ [Exemple avec pg_fetch_result](#)
- ◇ [Exemple avec pg_fetch_row](#)
- ◇ [Exemple avec pg_field_is_null](#)
- ◇ [Exemple avec pg_free_result](#)
- ◇ [Exemple avec pg_get_notify](#)
- ◇ [Exemple avec pg_get_pid](#)
- ◇ [Exemple avec pg_get_result](#)
- ◇ [Exemple avec pg_host](#)
- ◇ [Exemple avec pg_insert](#)

- ◇ [Exemple avec pg_last_error](#)
- ◇ [Exemple avec pg_last_error](#)
- ◇ [Exemple avec pg_last_oid](#)
- ◇ [Exemple avec pg_lo_close](#)
- ◇ [Exemple avec pg_lo_create](#)
- ◇ [Exemple avec pg_lo_export](#)
- ◇ [Exemple avec pg_lo_import](#)
- ◇ [Exemple avec pg_lo_open](#)
- ◇ [Exemple avec pg_lo_read](#)
- ◇ [Exemple avec pg_lo_read_all](#)
- ◇ [Exemple avec pg_lo_seek](#)
- ◇ [Exemple avec pg_lo_tell](#)
- ◇ [Exemple avec pg_lo_unlink](#)
- ◇ [Exemple avec pg_lo_write](#)
- ◇ [Exemple avec pg_num_fields](#)
- ◇ [Exemple avec pg_num_rows](#)
- ◇ [Exemple avec pg_options](#)
- ◇ [Exemple avec pg_parameter_status](#)
- ◇ [Exemple avec pg_pconnect](#)
- ◇ [Exemple avec pg_ping](#)
- ◇ [Exemple avec pg_port](#)
- ◇ [Exemple avec pg_prepare](#)
- ◇ [Exemple avec pg_put_line](#)
- ◇ [Exemple avec pg_result_error](#)
- ◇ [Exemple avec pg_result_error_field](#)
- ◇ [Exemple avec pg_result_seek](#)
- ◇ [Exemple avec pg_result_status](#)
- ◇ [Exemple avec pg_select](#)
- ◇ [Exemple avec pg_send_execute](#)
- ◇ [Exemple avec pg_send_query](#)
- ◇ [Exemple avec pg_send_query_params](#)
- ◇ [Exemple avec pg_set_client_encoding](#)
- ◇ [Exemple avec pg_set_error_verbosity](#)
- ◇ [Exemple avec pg_trace](#)
- ◇ [Exemple avec pg_transaction_status](#)
- ◇ [Exemple avec pg_tty](#)
- ◇ [Exemple avec pg_unescape_bytea](#)
- ◇ [Exemple avec pg_untrace](#)
- ◇ [Exemple avec pg_update](#)
- ◇ [Exemple avec pg_version](#)
- ◇ [Php comme module apache 1.3.x](#)
- ◇ [Php et apache 1.3.x en tant que cgi](#)
- ◇ [Php et apache 2.0.x en mode cgi](#)
- ◇ [Php et apache 2.0.x en tant que module](#)
- ◇ [Php:// et filtres d'écritures](#)
- ◇ [Exemple avec php_logo_guid](#)
- ◇ [Exemple avec php_sapi_name](#)
- ◇ [Exemple avec php_strip_whitespace](#)
- ◇ [Exemple avec phpinfo](#)
- ◇ [Exemple avec phpversion](#)
- ◇ [Exemple avec pi](#)
- ◇ [Positionnement dans un fichier](#)
- ◇ [Positionnement et suppression d'un verrou](#)
- ◇ [Exemple avec posix_access](#)

- ◇ [Exemple avec posix_getgid](#)
 - ◇ [Exemple avec posix_mknod](#)
 - ◇ [Exemple avec preg_grep](#)
 - ◇ [Exemple avec prev et ses amies](#)
 - ◇ [Exemple avec printer_abort](#)
 - ◇ [Exemple avec printer_close](#)
 - ◇ [Exemple avec printer_create_dc](#)
 - ◇ [Exemple avec printer_draw_bmp](#)
 - ◇ [Exemple avec printer_draw_chord](#)
 - ◇ [Exemple avec printer_draw_ellipse](#)
 - ◇ [Exemple avec printer_draw_line](#)
 - ◇ [Exemple avec printer_draw_pie](#)
 - ◇ [Exemple avec printer_draw_rectangle](#)
 - ◇ [Exemple avec printer_draw_roundrect](#)
 - ◇ [Exemple avec printer_draw_text](#)
 - ◇ [Exemple avec printer_get_option](#)
 - ◇ [Exemple avec printer_list](#)
 - ◇ [exemple avec printer_logical_fontheight](#)
 - ◇ [Exemple avec printer_open](#)
 - ◇ [exemple avec printer_select_brush](#)
 - ◇ [exemple avec printer_select_font](#)
 - ◇ [exemple avec printer_select_pen](#)
 - ◇ [Exemple avec printer_set_option](#)
 - ◇ [Exemple avec printer_start_doc](#)
 - ◇ [Exemple avec printer_write](#)
 - ◇ [Produit une liste de tous les services d'une société avec ldap_list](#)
 - ◇ [Profiler une section de code php](#)
 - ◇ [Exemple avec property_exists](#)
 - ◇ [Pré et post-incrémentation : exemples](#)
 - ◇ [Pré remplir un formulaire pdf](#)
 - ◇ [Prépare une requête sql avec des marqueurs](#)
 - ◇ [Présentation de dbm](#)
 - ◇ [Exemple avec pspell_add_to_personal](#)
 - ◇ [Exemple avec pspell_add_to_personal](#)
 - ◇ [Exemple avec pspell_check](#)
 - ◇ [Exemple avec pspell_config_create](#)
 - ◇ [Exemple avec pspell_config_ignore](#)
 - ◇ [Exemple avec pspell_config_mode](#)
 - ◇ [Exemple avec pspell_config_personal](#)
 - ◇ [Exemple avec pspell_config_repl](#)
 - ◇ [Exemple avec pspell_config_runtogether](#)
 - ◇ [Exemple avec pspell_new](#)
 - ◇ [Exemple avec pspell_new_config](#)
 - ◇ [Exemple avec pspell_new_personal](#)
 - ◇ [Exemple avec pspell_store_replacement](#)
 - ◇ [Exemple avec pspell_suggest](#)
- ◆ Q
- ◆ [Quelques exemples](#)
 - ◆ [Quelques exemples](#)

- ◇ [Quelques exemples](#)
- ◇ [Quelques exemples](#)
- ◇ [Quelques exemples](#)
- ◇ [Quelques exemples](#)
- ◇ [Quelques exemples avec pow](#)
- ◇ [Quelques exemples sur l'utilisation de l'option bindto](#)
- ◆ R
- ◆
 - ◇ [Exemple avec rad2deg](#)
 - ◇ [Exemple avec radius_acct_open](#)
 - ◇ [Exemple avec radius_add_server](#)
 - ◇ [Exemple avec radius_auth_open](#)
 - ◇ [Exemple avec radius_create_request](#)
 - ◇ [Exemple avec radius_cvt_addr](#)
 - ◇ [Exemple avec radius_cvt_int](#)
 - ◇ [Exemple avec radius_cvt_string](#)
 - ◇ [Exemple avec radius_get_attr](#)
 - ◇ [Exemple avec radius_get_vendor_attr](#)
 - ◇ [Exemple avec radius_put_attr](#)
 - ◇ [Exemple avec radius_put_int](#)
 - ◇ [Exemple avec radius_put_string](#)
 - ◇ [Exemple avec radius_put_vendor_attr](#)
 - ◇ [Exemple avec rand](#)
 - ◇ [Exemple avec range](#)
 - ◇ [Exemple avec rar::extract](#)
 - ◇ [Exemple avec rar::getattr](#)
 - ◇ [Exemple avec rar::getcrc](#)
 - ◇ [Exemple avec rar::getfiletime](#)
 - ◇ [Exemple avec rar::gethostos](#)
 - ◇ [Exemple avec rar::getmethod](#)
 - ◇ [Exemple avec rar::getname](#)
 - ◇ [Exemple avec rar::getpackedsized](#)
 - ◇ [Exemple avec rar::getunpackedsized](#)
 - ◇ [Exemple avec rar::getversion](#)
 - ◇ [Exemple avec rar_entry_get](#)
 - ◇ [Exemple avec rar_list](#)
 - ◇ [Exemple avec rawurldecode](#)
 - ◇ [Exemple avec rawurlencode](#)
 - ◇ [Exemple avec readline](#)
 - ◇ [Exemple avec readlink](#)
 - ◇ [Exemple avec realpath](#)
 - ◇ [Recherche avec ldap](#)
 - ◇ [Recherche ldap](#)
 - ◇ [Recherche les couples de balises html \(gourmand\)](#)
 - ◇ [Recherche parallèle utilisant yaz](#)
 - ◇ [Redimensionnement d'une image](#)
 - ◇ [Redimensionnement proportionnel d'une image](#)
 - ◇ [Refuser toutes les connexions depuis localhost](#)
 - ◇ [Exemple avec register_tick_function](#)
 - ◇ [Reinstaller apache avec pthread](#)
 - ◇ [Remplacer les url par des liens cliquables](#)
 - ◇ [Exemple avec rename](#)
 - ◇ [Exemple avec rename_function](#)
 - ◇ [Renommer un fichier via sftp](#)
 - ◇ [Reprise d'un chargement avec ftp_nb_put](#)

- ◇ [reprise d'un téléchargement à la position 100 dans un nouveau fichier avec ftp_nb_get](#)
- ◇ [require_once est insensible à la casse sur windows](#)
- ◇ [Requête d'effacement](#)
- ◇ [Requête valide](#)
- ◇ [Requêtes mysql](#)
- ◇ [Exemple avec reset](#)
- ◇ [Exemple avec restore_error_handler](#)
- ◇ [retirer une variable de session avec \\$ session et register_globals inactif.](#)
- ◇ [Retourne la première colonne de la ligne suivante](#)
- ◇ [Retourner des références](#)
- ◇ [Retourner une référence d'une fonction](#)
- ◇ [Exemple avec rotateto.php">swfdisplayitem->rotateto](#)
- ◇ [Rotation d'une image de 180 degrés](#)
- ◇ [Exemple avec round](#)
- ◇ [Exemple avec rsort](#)
- ◇ [Exemple avec rtrim](#)
- ◇ [Exemple avec runkit_class_adopt](#)
- ◇ [Exemple avec runkit_class_emancipate](#)
- ◇ [Exemple avec runkit_function_add](#)
- ◇ [Exemple avec runkit_function_copy](#)
- ◇ [Exemple avec runkit_function_redefine](#)
- ◇ [Exemple avec runkit_method_add](#)
- ◇ [Exemple avec runkit_method_copy](#)
- ◇ [Exemple avec runkit_method_redefine](#)
- ◇ [Exemple avec runkit_method_remove](#)
- ◇ [Exemple avec runkit_method_rename](#)
- ◇ [Exemple avec runkit_return_value_used](#)
- ◇ [Réception d'un fichier par méthode put](#)
- ◇ [Récupération d'informations des champs](#)
- ◇ [Récupération d'informations à propos des champs](#)
- ◇ [Récupération d'un attribut d'un noeud](#)
- ◇ [Récupération d'un code sqlstate](#)
- ◇ [Récupération d'un contenu](#)
- ◇ [Récupération de la liste des flags use](#)
- ◇ [Récupération de la valeur d'un attribut](#)
- ◇ [Récupération de multiples lignes de résultats retournées par une procédure stockée](#)
- ◇ [Récupération de tous les attributs d'un noeud](#)
- ◇ [Récupération de tous les livres anglais](#)
- ◇ [Récupération de tous les éléments xinclude](#)
- ◇ [Récupération des attributs de connexion à une base de données](#)
- ◇ [Récupération des informations du système depuis php](#)
- ◇ [Récupération des lignes avec un curseur scrollable](#)
- ◇ [Récupération des méta-informations pour une colonne](#)
- ◇ [Récupération des métadonnées d'une table](#)
- ◇ [Récupération du code php depuis un document html](#)
- ◇ [Récupération du noeud racine avec document-element.php">domdocument->document_element](#)
- ◇ [Récupération du nom du noeud](#)

- ◇ [Récupération du nom du type du document](#)
- ◇ [Récupération du nombre total de livres anglais](#)
- ◇ [Références dans un constructeur \(3\)](#)
- ◇ [Références sur les variables globales](#)
- ◇ [Résolution d'une équation diophantine linéaire](#)
- ◇ [Résoudre un nom de chemin](#)
- ◇ [Résultat de get_meta_tags](#)
- ◆ S
- ◆
 - ◇ [Sauvegarde d'un arbre dom dans un fichier](#)
 - ◇ [Sauvegarde d'un arbre html dans un fichier](#)
 - ◇ [Sauvegarde d'un arbre html dans une chaîne de caractères](#)
 - ◇ [Sauvegarde de l'arbre dom dans une chaîne de caractères](#)
 - ◇ [Sauver le résultat d'une transformation xslt dans un fichier](#)
 - ◇ [Sauver le résultat d'une transformation xslt dans une chaîne](#)
 - ◇ [Sauver/restaurer l'environnement cpdf](#)
 - ◇ [Scinde une chaîne et capture les positions](#)
 - ◇ [Exemple avec serialize](#)
 - ◇ [Exemple avec sesam_affected_rows](#)
 - ◇ [Exemple avec sesam_disconnect](#)
 - ◇ [Exemple avec sesam_errormsg](#)
 - ◇ [Exemple avec sesam_execimm](#)
 - ◇ [Exemple avec sesam_fetch_array](#)
 - ◇ [Exemple avec sesam_fetch_row](#)
 - ◇ [Exemple avec session_cache_expire](#)
 - ◇ [Exemple avec session_cache_limiter](#)
 - ◇ [Exemple avec session_name](#)
 - ◇ [Exemple avec session_regenerate_id](#)
 - ◇ [Exemple avec session_register](#)
 - ◇ [exemple avec session_set_save_handler](#)
 - ◇ [Exemple avec set_exception_handler](#)
 - ◇ [Exemple avec setcookie](#)
 - ◇ [Exemple avec setlocale pour windows](#)
 - ◇ [Exemple avec settype](#)
 - ◇ [Exemple avec sha1](#)
 - ◇ [Exemple avec shell_exec](#)
 - ◇ [Exemple avec shuffle](#)
 - ◇ [Exemple avec sin](#)
 - ◇ [Exemple avec sleep](#)
 - ◇ [Exemple avec sleep](#)
 - ◇ [Exemple avec snmp_get_quick_print](#)
 - ◇ [Exemple avec snmp_set_quick_print](#)
 - ◇ [Exemple avec snmpget](#)
 - ◇ [Exemple avec soapclient](#)
 - ◇ [Exemple avec soapclient->__getfunctions\(\)](#)
 - ◇ [Exemple avec soapclient->__getlastrequest\(\)](#)
 - ◇ [Exemple avec soapclient->__getlastresponse\(\)](#)
 - ◇ [Exemple avec soapclient->__gettypes\(\)](#)
 - ◇ [Exemple avec soapclient->__soapcall\(\)](#)
 - ◇ [Exemple avec socket_sendto](#)
 - ◇ [Exemple avec socket_set_block](#)
 - ◇ [Exemple avec socket_set_nonblock](#)
 - ◇ [Exemple avec socket_strerror](#)
 - ◇ [Exemple avec sort](#)
 - ◇ [Exemple avec spl_classes](#)

- ◇ [Exemple avec split](#)
- ◇ [Exemple avec spliti](#)
- ◇ [sprintf : notation scientifique](#)
- ◇ [spécifier une catégorie avec un format hiérarchisé](#)
- ◇ [Exemple avec sql_regcase](#)
- ◇ [Exemple avec sqlite_close](#)
- ◇ [Exemple avec sqlite_factory](#)
- ◇ [Exemple avec sqlite_fetch_single](#)
- ◇ [Exemple avec sqlite_open](#)
- ◇ [Exemple avec sqrt](#)
- ◇ [Exemple avec srand](#)
- ◇ [Exemple avec sscanf : utilisation des paramètres optionnels](#)
- ◇ [Statue un lien symbolique via sftp](#)
- ◇ [Statut d'un fichier via sftp](#)
- ◇ [Stocker des données sur un serveur distant](#)
- ◇ [Exemple avec str_ireplace](#)
- ◇ [Exemple avec str_pad](#)
- ◇ [Exemple avec str_repeat](#)
- ◇ [Exemple avec str_replace](#)
- ◇ [Exemple avec str_rot13](#)
- ◇ [Exemple avec str_word_count](#)
- ◇ [Exemple avec strcasecmp](#)
- ◇ [Exemple avec stream_context_create](#)
- ◇ [Exemple avec stream_context_get_default](#)
- ◇ [Exemple avec stream_copy_to_stream](#)
- ◇ [Exemple avec stream_get_contents](#)
- ◇ [Exemple avec stream_get_filters](#)
- ◇ [Exemple avec stream_get_transports](#)
- ◇ [Exemple avec stream_set_timeout](#)
- ◇ [Exemple avec stream_set_write_buffer](#)
- ◇ [Exemple avec stream_socket_enable_crypto](#)
- ◇ [Exemple avec stream_socket_pair](#)
- ◇ [Exemple avec stream_socket_recvfrom](#)
- ◇ [Exemple avec stream_socket_sendto](#)
- ◇ [Exemple avec streammp3.php">swfmovie->streammp3](#)
- ◇ [Exemple avec strip_tags](#)
- ◇ [Exemple avec strlen](#)
- ◇ [Exemple avec strnatcmp](#)
- ◇ [Exemple avec strpbrk](#)
- ◇ [Exemple avec strpos](#)
- ◇ [Exemple avec strptime](#)
- ◇ [Exemple avec strrchr](#)
- ◇ [Exemple avec strrpos](#)
- ◇ [Exemple avec strstr](#)
- ◇ [Exemple avec strtolower](#)
- ◇ [Exemple avec strtoupper](#)
- ◇ [Structure d'instruction exécutable](#)
- ◇ [Structure du paquet php 5](#)
- ◇ [Structure du tableau retourné par ldap_get_entries](#)
- ◇ [Style orienté objet](#)

- ◇ [Style procédural](#)
- ◇ [Substitution des xincludes](#)
- ◇ [Exemple avec substr_compare](#)
- ◇ [Exemple avec substr_count](#)
- ◇ [Exemple avec substr_replace](#)
- ◇ [Superglobales particulières avec runkit.superglobal=foo_bar dans php.ini](#)
- ◇ [Suppression d'un attribut](#)
- ◇ [Suppression de caractères dans un tableau avec trim](#)
- ◇ [Suppression de commentaires avec le tokenizer](#)
- ◇ [Supprimer un noeud en dom xml](#)
- ◇ [Exemple avec swf_addbuttonrecord](#)
- ◇ [Exemple avec swfaction](#)
- ◇ [Exemple avec swfbitmap](#)
- ◇ [Exemple avec swfgradient](#)
- ◇ [Exemple avec swfmorph](#)
- ◇ [Exemple avec swfshape](#)
- ◇ [Exemple avec swftext](#)
- ◇ [Exemple avec sybase_connect](#)
- ◇ [sybase_fetch_object retourne l'objet foo](#)
- ◇ [Exemple avec sybase_unbuffered_query](#)
- ◇ [Syntaxe complexe de chaîne](#)
- ◇ [Exemple avec system](#)
- ◇ [Séparation des instructions](#)
- ◆ T
- ◆
 - ◇ [Exemple avec tan](#)
 - ◇ [Exemple avec tempnam](#)
 - ◇ [Test d'existence d'un fichier](#)
 - ◇ [Test du support exslt](#)
 - ◇ [Test votre implémentation dom](#)
 - ◇ [Teste l'existence d'un attribut](#)
 - ◇ [Exemple avec tidy::construct](#)
 - ◇ [Exemple avec tidy_access_count](#)
 - ◇ [Exemple avec tidy_clean_repair](#)
 - ◇ [Exemple avec tidy_config_count](#)
 - ◇ [Exemple avec tidy_diagnose](#)
 - ◇ [Exemple avec tidy_error_count](#)
 - ◇ [Exemple avec tidy_get_body](#)
 - ◇ [Exemple avec tidy_get_config](#)
 - ◇ [Exemple avec tidy_get_error_buffer](#)
 - ◇ [Exemple avec tidy_get_head](#)
 - ◇ [Exemple avec tidy_get_html](#)
 - ◇ [Exemple avec tidy_get_output](#)
 - ◇ [Exemple avec tidy_get_status](#)
 - ◇ [Exemple avec tidy_getopt](#)
 - ◇ [Exemple avec tidy_parse_file](#)

- ◇ [Exemple avec tidy_parse_string](#)
- ◇ [Exemple avec tidy_repair_file](#)
- ◇ [Exemple avec tidy_repair_string](#)
- ◇ [Exemple avec tidy_setopt](#)
- ◇ [Exemple avec tidy_warning_count](#)
- ◇ [Exemple avec time](#)
- ◇ [Exemple avec time](#)
- ◇ [Exemple avec time_nanosleep](#)
- ◇ [Exemple avec time_sleep_until](#)
- ◇ [Exemple avec tmpfile](#)
- ◇ [Exemple avec token_get_all](#)
- ◇ [Exemple avec token_name](#)
- ◇ [Exemple avec touch](#)
- ◇ [Tracer une ligne fine](#)
- ◇ [Transcription des comparaisons standards des tableaux](#)
- ◇ [Transformation d'une chaîne](#)
- ◇ [Transformation en un document dom](#)
- ◇ [Transformation en un fichier html](#)
- ◇ [Transtypage](#)
- ◇ [Transtypage en chaîne](#)
- ◇ [Transtypage xml -> html](#)
- ◇ [Travail avec marcxml](#)
- ◇ [Traverser un pseudo-tableau children\(\)](#)
- ◇ [tri avec usort sur un objet](#)
- ◇ [Tri non sensible à la casse](#)
- ◇ [Exemple avec trigger_error](#)
- ◇ [Trouver le document d'un noeud avec dom xml](#)
- ◇ [Téléchargement d'un fichier via scp](#)
- ◇ [Télécharger plusieurs fichiers simultanément](#)

◆ U

- ◆ [Exemple avec uasort](#)
- ◆ [Exemple avec ucfirst](#)
- ◆ [Exemple avec ucwords](#)
- ◆ [Exemple avec udm_api_version](#)
- ◆ [Exemple avec udm_cat_list](#)
- ◆ [Exemple avec udm_load_ispell_data et udm_ispell_type_server](#)
- ◆ [Exemple avec uksort](#)
- ◆ [Exemple avec umask](#)
- ◆ [Exemple avec un destructeur](#)
- ◆ [un exemple de liste retournée par php_ini_scanned_files](#)
- ◆ [Un exemple simple avec stripslashes](#)
- ◆ [un moyen pratique pour remplacer opendir par glob](#)
- ◆ [Une classe de flux pour accéder aux variables globales](#)
- ◆ [une comparaison simple empty / isset .](#)
- ◆ [Exemple avec une méthode statique](#)
- ◆ [Une navigation de fiches plus sécuritaire](#)
- ◆ [Exemple avec uniqid](#)
- ◆ [Exemple avec unpack](#)
- ◆ [Exemple avec unserialize](#)
- ◆ [Exemple avec unset et une variable statique](#)
- ◆ [Exemple avec urldecode](#)
- ◆ [exemple avec urlencode et htmlentities](#)
- ◆ [Using pg_query_params](#)
- ◆ [Utilisation basique de l'api de réflexion](#)

- ◇ Utilisation d'une "chaîne de caractères" en chiffre
- ◇ Utilisation d'une connexion udp
- ◇ Utilisation d'une constante dans un index de tableau(2)
- ◇ Utilisation d'une image pour soumettre un formulaire
- ◇ Utilisation dangereuse de variables
- ◇ Utilisation de crypt avec différents types de cryptage
- ◇ Utilisation de ftp_raw pour s'identifier sur un serveur ftp manuellement.
- ◇ Utilisation de gd_info
- ◇ utilisation de header pour générer un fichier de type pdf ou d'un autre type
- ◇ Utilisation de http_build_query avec un objet
- ◇ Utilisation de mcrypt_module_open pour encrypter
- ◇ Utilisation de pg_query avec plusieurs requêtes
- ◇ Utilisation de socket_bind pour définir l'adresse de la source
- ◇ Utilisation de ssh2_auth_none pour récupérer la liste des méthodes d'identification.
- ◇ utilisation de stripslashes sur un tableau
- ◇ Utilisation de syslog
- ◇ utilisation de asxml() avec les résultats de xpath
- ◇ Utilisation de base de tidy
- ◇ Utilisation de connexions udp
- ◇ Utilisation de curl pour récupérer une page
- ◇ Utilisation de fonctions anonymes comme fonction de callback
- ◇ Utilisation de l'enregistrement de messages xslt
- ◇ Utilisation de l'opérateur instanceof avec php 5
- ◇ Utilisation de la classe reflectionclass
- ◇ Utilisation de la classe reflectionextension
- ◇ Utilisation de la classe reflectionfunction
- ◇ utilisation de la classe reflectionmethod
- ◇ Utilisation de la classe reflectionparameter
- ◇ utilisation de la classe reflectionproperty
- ◇ Utilisation de la sortie du buffer pour inclure un fichier php dans une chaîne
- ◇ Utilisation de paquets incrémentaux
- ◇ Utilisation de pdo::attr_driver_name
- ◇ Utilisation de procédures stockées
- ◇ Utilisation de ptys
- ◇ Utilisation de sleep et de wakeup
- ◇ Utilisation de votre propre gestionnaire d'erreurs comme méthode
- ◇ Utilisation des accolades {} dans les chaînes pour les tableaux
- ◇ utilisation des constantes true et false avec une structure de contrôle
- ◇ Utilisation des fichiers binaires avec fread
- ◇ Utilisation des tableaux avec setcookie
- ◇ Utilisation du paramètre count
- ◇ Utilisation du paramètre return de print_r
- ◇ Utiliser fpassthru avec un fichier binaire
- ◇ Utiliser des xforms pour remplir \$_post
- ◇ Utiliser le type html pour les extensions php
- ◇ Utiliser un mot de passe et md5
- ◇ Utiliser un ref cursor issu d'une commande select
- ◇ Utiliser un serveur de socket udp
- ◇ utiliser une valeur négative avec length
- ◆ V
- ◆ ◇ Valeurs par défaut des variables non initialisées
- ◇ Validation d'adresse ip
- ◇ Valide une transaction

- ◇ [Valider une transaction sesam](#)
- ◇ [Exemple avec var_dump](#)
- ◇ [Variables de formulaires complexes](#)
- ◇ [Exemple avec version_compare](#)
- ◇ [vfprintf : zéro comme caractères d'espacements](#)
- ◇ [Vide une base de données dbase](#)
- ◇ [Vérification gettext](#)
- ◇ [Vérification d'erreur](#)
- ◇ [Vérification de l'existence d'un gestionnaire de flux](#)
- ◇ [Vérification de la version du protocole avec ldap_get_option](#)
- ◇ [Vérification renforcée de noms de fichiers](#)
- ◇ [Vérifie si l'administrateur système a signé le fichier](#)
- ◇ [vérifier la présence de constantes avec defined](#)
- ◇ [Vérifier le nom d'une couleur avec un masque shell](#)
- ◆ **W**
- ◆
 - ◇ [wddx_serialize_vars](#)
 - ◇ [Exemple avec while](#)
 - ◇ [Exemple avec wordwrap avec césure](#)
- ◆ **X**
- ◆
 - ◇ [Exemple avec xattr_supported](#)
 - ◇ [Exemple avec xdiff_file_diff](#)
 - ◇ [Exemple avec xdiff_file_diff_binary](#)
 - ◇ [Exemple avec xdiff_file_merge3](#)
 - ◇ [Exemple avec xdiff_file_patch_binary](#)
 - ◇ [Exemple avec xdiff_string_diff](#)
 - ◇ [Exemple avec xdiff_string_patch](#)
 - ◇ [Exemple avec xml_set_object](#)
 - ◇ [Xmltest2.xml](#)
 - ◇ [XPath](#)
 - ◇ [Exemple avec xpath_eval_expression](#)
 - ◇ [Exemple avec xslt_backend_name](#)
 - ◇ [Exemple avec xslt_backend_version](#)
 - ◇ [Exemple avec xslt_create](#)
 - ◇ [Exemple avec xslt_set_error_handler](#)
 - ◇ [Exemple avec xslt_setopt](#)
- ◆ **Y**
- ◆
 - ◇ [Exemple avec yp_err_string](#)
 - ◇ [Exemple avec yp_first](#)
 - ◇ [Exemple avec yp_match](#)
 - ◇ [Exemple avec yp_next](#)
 - ◇ [Exemple avec yp_order](#)
- ◆ **Z**
- ◆
 - ◇ [Exemple avec zend_logo_guid](#)
 - ◇ [Exemple avec zend_version](#)